This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LeanVAE: An Ultra-Efficient Reconstruction VAE for Video Diffusion Models

Yu Cheng[1,2]
[1]Zhejiang University
chengyu@westlake.edu.cn

Fajie Yuan[2*]
[2]Westlake University
yuanfajie@westlake.edu.cn

Abstract

*Recent advances in Latent Video Diffusion Models (LVDMs) have revolutionized video generation by leveraging Video Variational Autoencoders (Video VAEs) to compress intricate video data into a compact latent space. However, as LVDM training scales, the computational overhead of Video VAEs becomes a critical bottleneck, particularly for encoding high-resolution videos. To address this, we propose **LeanVAE**, a novel and ultra-efficient Video VAE framework that introduces two key innovations: (1) a lightweight architecture based on a Neighborhood-Aware Feedforward (NAF) module and non-overlapping patch operations, drastically reducing computational cost, and (2) the integration of wavelet transforms and compressed sensing techniques to enhance reconstruction quality. Extensive experiments validate LeanVAE's superiority in video reconstruction and generation, particularly in enhancing efficiency over existing Video VAEs. Our model offers up to 50× fewer FLOPs and 44× faster inference speed [1] while maintaining competitive reconstruction quality, providing insights for scalable, efficient video generation. Our models and code are available at https://github.com/westlake-repl/LeanVAE.*

1. Introduction

Video generation models have garnered widespread attention due to their significant impact on fields such as animation, advertisement, and simulation [5]. Among these, Latent Video Diffusion Models (LVDMs) have emerged as a leading approach. In these models, Video Variational Autoencoders (Video VAEs) facilitate a bidirectional mapping between the high-dimensional, redundant video pixel space and a low-dimensional, compact latent space, enabling both spatial and temporal compression of the 3D RGB volume. A diffusion denoising model is then trained to model the distribution of the latent space. This latent space design offers substantial advantages in training efficiency, stability, and scalability compared to pixel-space diffusion models, leading to the widespread adoption of LVDMs in notable works such as Open-Sora-Plan [19], Open-Sora [48], Latte [23], CogVideoX [42], EasyAnimate [40], HunyuanVideo [36], and Cosmos [1].

The Video VAEs play a crucial role in LVDM models. The quality of the video reconstruction directly determines the upper bound of the generated video quality in LVDMs. Artifacts [9] such as blurring, distortions, and motion inconsistencies in VAE outputs directly degrade the final generation. Besides, the encoding efficiency of the VAE often becomes a computational bottleneck during LVDM training [22], particularly for long-duration, high-resolution videos. Therefore, optimizing the Video VAE is essential to enable efficient large-scale pre-training for LVDMs.

Given the significant value of an efficient and powerful Video VAE in LVDMs, many research has been devoted to this area. Early works like OD-VAE [8], CV-VAE [47], and SliceVAE [40] typically extended the well-known Stable Diffusion image VAE (SD VAE) [26] by inflating 2D convolutions to 3D and incorporating spatiotemporal attention modules to achieve temporal and spatial compression for videos. While these models demonstrated promising results, they incur prohibitive computational costs. For example, OD-VAE consumes around 32GB of memory to encode and decode a 5-frame 1080 square video with FP16 inference. Subsequent research has pursued two main strategies to address the computational burden. In works like Open-Sora, Movie Gen [24], Cosmos Tokenizer [1], and VidTok [30], various techniques have been proposed to factorize the computationally intensive 3D structures. Other works focus on building simplified network architectures. For example, Cosmos Tokenizer [1], WF-VAE [22] reduce network redundancies by introducing wavelet transforms. Omnitokenizer [33], ViTok [12] build much smaller transformer-based models by leveraging global attention, though they suffer from quadratic computational complexity with respect to video resolution.

Inspired by above works, we aim to advance Video VAE and demonstrate a significant improvement in balancing efficiency and performance by proposing a novel and highly efficient Video VAE approach, coined LeanVAE. Unlike

*Corresponding author.
[1]Evaluated on 768 × 768 videos.

previous methods that either inherited the Video VAE from the SD VAE or employed ViT [3] architectures, we use two key designs to ensure a lightweight network. First, we employ a non-overlapping patch operation to downsample both spatial and temporal dimensions at the very beginning of the network, drastically reducing the computation. Second, we introduce a Neighborhood-Aware Feedforward (NAF) module as the backbone of our model, as shown in Fig. 1(b). The NAF module efficiently captures local contexts via depthwise convolutions and processes them through a feedforward network, enabling lightweight yet powerful extraction and transformation of video features.

After that, we integrate wavelet transform [6] and Compressed Sensing (CS) [15] technique to further enhance reconstruction quality. Specifically, we apply wavelet transforms to the input RGB signal following prior work [22, 27], obtaining features in the frequency domain that enrich the input space and aid in expanding the model's receptive field [16]. Another key improvement is the adoption of a classical CS algorithm for latent channel dimension compression. This addresses the inefficiency we identified in conventional two-linear-layer AutoEncoding (AE) [15] structure for this task. To the best of our knowledge, this is the first application of CS for channel compression bottleneck in Video VAEs. Our experiments demonstrate a significant performance boost, revealing the immense potential of CS in this domain.

Finally, we explore the design space of LeanVAE and propose several variations for further optimization [27]. Through careful experimentation, we identify the optimal configuration and train a series of highly efficient LeanVAE models. We then demonstrate its superiority in video reconstruction against various strong baselines and highlight its strengths in both video reconstruction and generation. Fig. 3 and Appendix B illustrate its ability to achieve high reconstruction quality while significantly reducing computational cost, with this advantage especially pronounced at higher resolutions. It is also worth mentioning that LeanVAE supports joint modeling of images and videos and maintains causal property of the latent space, making it broadly applicable. Additionally, we identify that the common practice in ViT of applying LayerNorm [4] to pixel patches is a key factor contributing to block artifacts [20] in reconstruction. We believe this is an interesting discovery that may inspire improvements in similar models for low-level vision tasks.

In summary, we present a highly efficient and lightweight video reconstruction framework that accelerates Video VAE and improves training throughput for LVDMs. We believe this approach marks a important enhancement toward more scalable and efficient video generation. There are three technical novelty that contributed to the good performance and merits of LeanVAE:

- We adopt the patch method inspired by ViT and design a lightweight NAF module as the model backbone, achieving a ultra-efficient network architecture.
- We enrich input representations using wavelet transforms and, for the first time, introduce a CS framework as the channel compression bottleneck in Video VAE. Both techniques significantly enhance model performance.
- We conduct an in-depth architectural ablation for further optimization. Extensive experiments validate the superiority of LeanVAE in both video reconstruction and generation, particularly in enhancing efficiency over existing Video VAEs.

2. Related Work

VAEs are powerful models for learning high-dimensional data distributions in either continuous or discrete [10, 32, 44] latent space. This work focuses on continuous VAEs, which are widely used in latent diffusion models to reduce the computational cost. We categorize related work into two main directions: Regular Video VAEs and Specific Video VAEs.

Regular Video VAEs Regular Video VAEs compress the time, height, and width dimensions of videos into 3D latent representations that resemble visual thumbnails. These models typically achieve a 4×8×8 compression ratio for the original videos and are widely adopted in popular open-source video generation frameworks [1, 19, 40, 42, 48]. To construct a Video VAE, the most straightforward idea is to inherit from the well-established SD image VAE to leverage its spatial compression priors. This involves inflating 2D convolutions in the image model to 3D convolutions as the video counterpart. Building on this, various techniques have been introduced to improve reconstruction quality. Pioneering works like OD-VAE [8], CV-VAE [47], and SliceVAE [40] introduced attention layers to model global spatiotemporal dependencies. Models like IV-VAE [37], VideoVAE+ [38], and FILM-VAE [2] proposed well-curated spatiotemporal convolutional modules for smoother motion and enhanced spatial details. CogVideoX [42] introduced increasing the latent channel dimension for better reconstruction. These innovations have effectively pushed the boundaries of reconstruction quality.

However, as video diffusion models scale, the computational bottleneck caused by Video VAE models in the LVDM pipeline has become increasingly unaffordable, leading to growing interest in a better balance between efficiency and reconstruction performance. Some works have attempted to decompose the 3D network structure required for spatiotemporal compression to reduce computation. In Open-Sora [48] and MovieGen [24], 2D+1D network architectures are chosen to compress the spatial dimensions first, followed by temporal compression. Cosmos Tokenizer [1] replaced standard 3D convolutions with

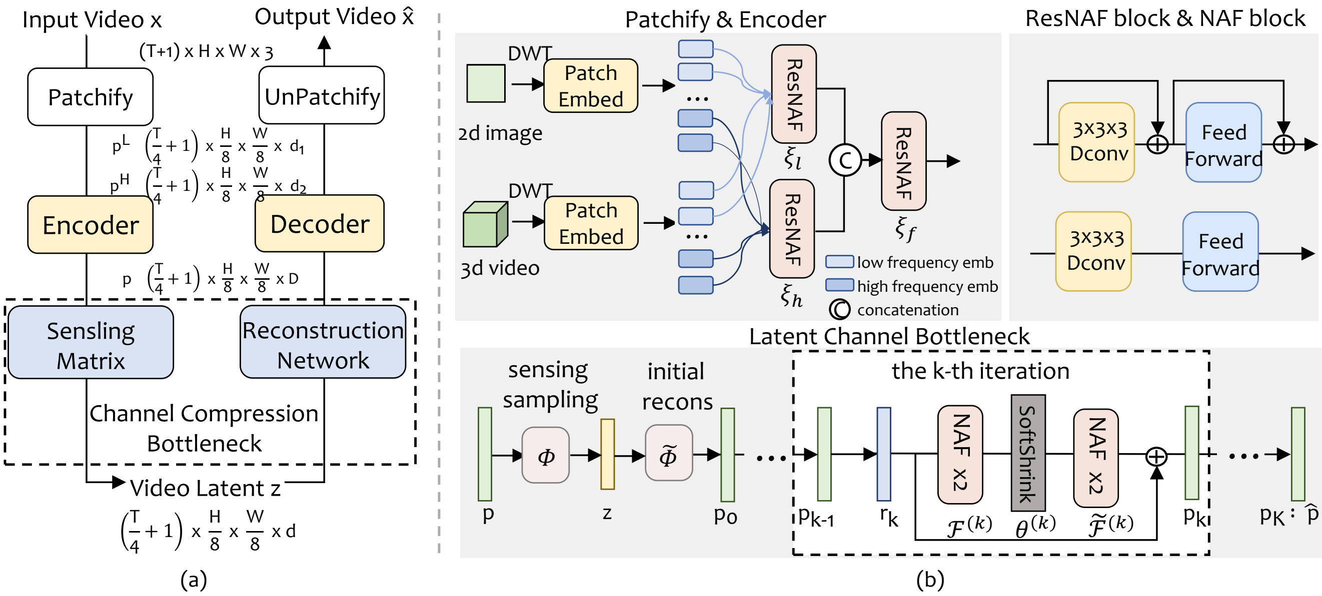

Figure 1. (a) LeanVAE framework overview. (b) Key components: **Patchifier** for image-video joint patching in frequency domain; ***Encoder*** for hierarchical feature extraction; **(Res)NAF** serves as model backbone, enabling Neighborhood-Aware Feedforward (with Residual connections); **Latent Channel Bottleneck** for latent channel compression and restoration based on *ISTA-Net*$^+$ algorithm.

factorized convolutions, applying 2D spatial convolutions followed by a temporal convolution. VidTok [30] further incorporates AlphaBlender into this factorized structure for better temporal modeling. Other works focus on approaches to simplify the network architecture. In Cosmos Tokenizer [1] and WF-VAE [22], wavelet transforms are applied to videos, resulting in lighter VAE models that reduce computation while maintaining good reconstruction quality. Transformer-based methods like OmniTokenizer [33], and ViTok [12] leveraged global dependency modeling to build models with much fewer parameters, but they suffer quadratic complexity with respect to video resolution. Building on these insights, our work achieves an excellent balance between efficiency and quality via a carefully designed lightweight network architecture, alongside clever use of wavelet transforms and compressive sensing.

Specific Video VAEs In LVDM, some works focus on training VAEs to generate specialized video latent spaces that enable more efficient training of diffusion models. For instance, HVDM [16] uses a hybrid video autoencoder to obtain a latent space that combines 2D triplane representations with 3D volume information. VidTwin [34] incorporates carefully designed submodules to disentangle the structure and dynamics in the latent space. LTX-Video [11] proposes a video compression method that increases spatiotemporal compression ratios and latent channel dimension simultaneously so as to relocate the patchifying operation from the diffusion model to the VAEs. These works require designing corresponding diffusion models to accommodate the unique characteristics of the latent space. Due to differences in compression ratios and latent space properties compared to regular Video VAEs, we consider these works as future directions for research rather than baselines for comparison.

3. Preliminary

In this section, we briefly introduce two key techniques utilized in LeanVAE: Wavelet Transform and Compressed Sensing.

3.1. Wavelet Transform

Wavelet transform is widely used in fields such as medical imaging and visual compression [27]. It decomposes visual signals in the RGB space into multiple frequency-domain components. In our work, we use the Haar wavelet transform to process images and videos, which includes a pair of reversible processes: Discrete Wavelet Transform (DWT) and Inverse Discrete Wavelet Transform (IDWT). For a 2D image $\mathbf{I} \in \mathbb{R}^{c\times h\times w}$, where c, h, and w represent channels, height, and width, respectively, the Haar 2D kernels operate along the height and width axes, decomposing the image into four subbands: $\{\mathbf{I}_{ll}, \mathbf{I}_{lh}, \mathbf{I}_{hl}, \mathbf{I}_{hh}\}$, each of size $c \times \frac{h}{2} \times \frac{w}{2}$. Similarly, given a 3D video $\mathbf{V} \in \mathbb{R}^{c\times t\times h\times w}$, where t denotes the temporal frames, the 3D Haar DWT is performed, dividing the input volume into eight subbands: $\{\mathbf{V}_{lll}, \mathbf{V}_{llh}, \mathbf{V}_{lhl}, \mathbf{V}_{lhh}, \mathbf{V}_{hll}, \mathbf{V}_{hlh}, \mathbf{V}_{hhl}, \mathbf{V}_{hhh}\}$, each of size $c \times \frac{t}{2} \times \frac{h}{2} \times \frac{w}{2}$. Among these subbands, the first component $(\mathbf{I}_{ll} or \mathbf{V}_{lll})$ represents low-frequency components providing long-term average information. The remaining components capture high-frequency details and local changes. These subbands can be perfectly reconstructed into the original signal using IDWT, as presented in the Appendix Fig. 6. After applying the Haar DWT, the spatiotemporal dimensions of the input visual signal are halved, while the channel dimension increases accordingly. This enriches the feature space and has been shown to effectively expand the receptive field of CNNs [16].

3.2. Compressed Sensing

Compressed sensing is a promising technique that enables signal reconstruction with far fewer measurement data than required by the Nyquist-Shannon sampling theorem [45]. Over the past decade, numerous effective CS methods have been developed for signal compression and restoration, which expresses the original signal with a reduced number of measurements and restores the signal through a recovery algorithm. For a signal $\mathbf{p} \in \mathbb{R}^D$, sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{d \times D}$, the measurements $\mathbf{z} \in \mathbb{R}^d (d \ll D)$ is obtained by the sampling process:

$$\mathbf{z} = \mathbf{\Phi}\mathbf{p}.$$

To achieve reliable reconstruction $\hat{\mathbf{p}}$ from reduced measurements $\mathbf{z}$, CS methods typically solve the following optimization problem:

$$\underset{\mathbf{p}}{\mathbf{argmin}} \ \frac{1}{2}\|\mathbf{z} - \mathbf{\Phi}\mathbf{p}\|_2^2 + \lambda\|\mathbf{\Psi}\mathbf{p}\|_1, \tag{1}$$

where $\|\mathbf{z} - \mathbf{\Phi}\mathbf{p}\|_2^2$ ensures data fidelity and $\|\mathbf{\Psi}\mathbf{p}\|_1$ promotes sparsity in $\mathbf{\Psi}\mathbf{p}$, with λ as the regularization parameter.

In this work, we use CS algorithm to build the latent channel bottleneck, which performs downsampling and reconstruction of the latent channels. To be specific, we employ *ISTA-Net*$^+$ [45], a unfolding algorithm framework that effectively optimizes Eq. (1) through K iterative steps. In the k-*th* iteration, it sequentially updates the intermediate reconstruction result $\mathbf{r}^{(k)}$ and final reconstruction $\mathbf{p}^{(k)}$ as follows:

$$\mathbf{r}^{(k)} = \mathbf{p}^{(k-1)} - \rho^{(k)}\mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{p}^{(k-1)} - \mathbf{z}). \tag{2}$$

$$\mathbf{p}^{(k)} = \mathbf{r}^{(k)} + \widetilde{\mathcal{F}}^{(k)}(soft(\mathcal{F}^{(k)}(\mathbf{r}^{(k)}), \theta^{(k)})). \tag{3}$$

The reconstruction network is parameterized by $\mathbf{\Theta} = \{\rho^{(k)}, \theta^{(k)}, \mathcal{F}^{(k)}, \widetilde{\mathcal{F}}^{(k)}\}_{k=1}^K$, where $\mathcal{F}^{(k)}$ and $\widetilde{\mathcal{F}}^{(k)}$ represent the forward and backward networks at the k-th iteration, respectively, and $\rho^{(k)}$ and $\theta^{(k)}$ are scalar parameters. The term $soft$ refers to the soft-shrinkage threshold operation:

$$soft(\boldsymbol{x}, \theta) = \mathrm{sgn}(\boldsymbol{x}) \cdot \max(|\boldsymbol{x}| - \theta, 0) \tag{4}$$

4. Method

4.1. Overall Process

Let the input video be denoted as $\mathbf{x} \in \mathbb{R}^{(T+1)\times H\times W\times 3}$, where (1 + T) is the number of frames (with T = 0 for images) and H × W denotes the spatial resolution. As shown on the left side of Fig. 1(a), the input video $\mathbf{x}$ is encoded into the compressed latent $\mathbf{z} \in \mathbb{R}^{(T'+1)\times H'\times W'\times d}$,with a spatial compression factor of $c_t = \frac{T}{T'}$, a temporal compression factor of $c_s = \frac{H}{H'} = \frac{W}{W'}$, and d is the latent channel dimension. The video is first processed with a patchify operation to obtain patch embeddings, including high-frequency component (HC) embedding $\mathbf{p^H} \in \mathbb{R}^{(T'+1)\times H'\times W'\times d_1}$ and low-frequency component (LC) embedding $\mathbf{p^L} \in \mathbb{R}^{(T'+1)\times H'\times W'\times d_2}$. These embeddings are then passed through the *Encoder* to facilitate feature interaction and transformation, producing the feature tensor $\mathbf{p} \in \mathbb{R}^{(T'+1)\times H'\times W'\times D}$. Finally, a sensing matrix reduces the dimension of $\mathbf{p}$ from D to d, resulting in the video latent $\mathbf{z}$. The right side of Fig. 1(a) depicts the inverse process, where $\mathbf{z}$ is decoded back into the reconstructed video $\hat{\mathbf{x}}$. We set $c_s = 8, c_t = 4, d_1 = 128, d_2 = 384, D = 512$ and $d \in 4, 16$. Detailed descriptions of each component are provided in the following sections.

4.2. Architecture Details

Patchify and UnPatchify. For the given input $\mathbf{x} \in \mathbb{R}^{(T+1)\times H\times W\times 3}$, we process the first frame $\mathbf{x_0} \in \mathbb{R}^{1\times H\times W\times 3}$ and following frames $\mathbf{x_{1:T}} \in \mathbb{R}^{T\times H\times W\times 3}$ separately for the joint encoding of videos and images. Specifically, for the continuous frames $\mathbf{x_{1:T}}$, we apply a 3D wavelet transform to map them into LC coefficients (the first subband) and HC coefficients (the remaining subbands concatenated along the channel dimension). The shape of these components are $\frac{T}{2}\times\frac{H}{2}\times\frac{W}{2}\times 3$ for LC and $\frac{T}{2}\times\frac{H}{2}\times\frac{W}{2}\times 21$ for HC. These features are then split into non-overlapping patches, with a patch size of $2 \times 4 \times 4$. After that, we project the LC and HC patches with two linear layers to obtain their patch embeddings $\mathbf{p^L_{1:T}}$ and $\mathbf{p^H_{1:T}}$. Similarly, the first frame $\mathbf{x_0}$ undergoes a 2D Haar transform and is divided into 4×4 patches to obtain the corresponding low-frequency and high-frequency patch vectors $\mathbf{p^L_0}$ and $\mathbf{p^H_0}$. The $\mathbf{p^L_0}$ and $\mathbf{p^L_{1:T}}$ are then concatenated along the temporal dimension as the LC embedding $\mathbf{p^L}$, and the same process is applied to the HC embeddings to form $\mathbf{p^H}$. The unpatchify process mirrors the patchify operation. We use the linear layer to project the patch embeddings back to the frequency domain coefficients, followed by resizing and applying the IDWT to reconstruct the RGB visual output $\hat{\mathbf{x}}$.

Notably, our approach distinguishes from traditional vision transformer patching in three points: 1) we apply the separate patch operation for the first frame and the remaining frames, enabling our model to handle both images and videos. 2) We enrich the representation space by operating in the frequency domain rather than the traditional RGB space. 3) We do not apply normalization to visual patches. We observed it degrades the performance and may result in block artifacts reported in transformer models [20].

Encoder and Decoder. We build a lightweight *Encoder* network ξ to further process $\mathbf{p^L}$ and $\mathbf{p^H}$, enabling feature interaction and transformation to obtain the final feature vector $\mathbf{p}$. Based on the ablation studies of the model architecture (See Sec. 5.4), we adopt a structure that first processes the low- and high-frequency features separately before fusing them. The ξ consist of three modules, that is ξ_l to process $\mathbf{p^L}$, ξ_h to process $\mathbf{p^H}$, and ξ_f for fusing two features. This process can be expressed as:

$$\mathbf{p} = \xi_f(cat(\xi_l(\mathbf{p^L}), \xi_h(\mathbf{p^H}))) \tag{5}$$

where cat denotes concatenation along the channel dimension.

We use the Neighborhood-Aware Feedforward with Residual Connection (ResNAF) module as the backbone to build the network $\xi = \{\xi_l, \xi_h, \xi_f\}$, as shown in Fig. 1(b). In ResNAF, the 3D input vector first goes through a depthwise convolution with kernel size $3 \times 3 \times 3$ to efficiently aggregate local neighboring contexts. The contextual features are then further enriched and transformed through a feedforward layer. Meanwhile, residual connections are incorporated to optimize feature propagation. We construct the ξ_l, ξ_h, ξ_f by stacking 2, 2, and 4 layers of ResNAF, respectively. Through the hierarchical stacking of ResNAF modules, *Encoder* achieves effective information propagation and feature modeling. The *Decoder* ψ follows a symmetric structure to the *Encoder*.

Notably, we apply causal padding in the temporal dimension for depthwise convolution, ensuring that each frame only interacts with previous frames. This causal property offers several benefits: 1) It improves modeling for images [44], as they are independent of neighboring frames. 2) It supports downstream models operating on causal latent spaces [21], broadening the applicability of LeanVAE. 3) It enables efficient temporal tiling inference by caching frames from previous chunks, ensuring continuity in convolutional operations, see Appendix E.

Channel Compression Bottleneck. This module acts as a latent channel bottleneck, performing both downsampling and upsampling operations. The downsampling part compresses the feature $\mathbf{p}$ from channel dimension D to d, obtaining the final video latent $\mathbf{z}$, which is subsequently used by the diffusion model. The upsampling part maps the channel of $\mathbf{z}$ back to D, generating the feature $\hat{\mathbf{p}}$ that is fed into the *Decoder* ψ.

To implement this, we adopt the classical *ISTA-Net*$^+$ framework [45], which uses a sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{d\times D}$ for channel downsampling and a recovery algorithm for upsampling. In the recovery process, the latent $\mathbf{z}$ is first multipied by a learnable matrix $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{D\times d}$ to get the initial reconstruction $\mathbf{p_0}$. This is then refined by the recovery network $\mathbf{\Theta}$ to obtain final result $\hat{\mathbf{p}}$. The network $\mathbf{\Theta}$ is a deep unfolding network that simulates an iterative optimization process, with the computation method defined by Eq. (2) and Eq. (3) at each iteration. The entire computation process is illustrated in Fig. 1(b). We employ a two-layer NAF network to construct the forward network $\mathcal{F}^{(k)}$ and backward network $\widetilde{\mathcal{F}}^{(k)}$, with $\rho^{(k)}$ and $\theta^{(k)}$ set as learnable scalars. The number of iterations K is set to 2.

While previous work often employs a simple AE structure with two linear layers for this purpose, we find this approach suboptimal. Instead, we leverage the *ISTA-Net*$^+$ framework to achieve effective downsampling and upsampling for the latent channel. Experiments demonstrate its strong effectiveness, leading to significant performance improvements in our model. We provide a straightforward comparison between AE and CS signal recovery algorithms in Appendix D for further clarity.

4.3. Training Objective

The entire model is then trained end-to-end. Our loss function includes four components, including a L1 reconstruction loss computed in both the RGB and frequency domains, a perceptual loss [46]) based on VGG features, an adversarial loss computed using a PatchGAN from work [49], and a KL regularization term [18]. The final loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{KL}\mathcal{L}_{KL}. \tag{6}$$

5. Experiments

5.1. Experimental Setup

Training Details We train the model using Kinetics-600 dataset [7]. The videos are resized and cropped to resolutions ranging from 96×128 to 352×288, with their original aspect ratios preserved. For each training step, we randomly sample 17 consecutive frames from a video. First, we train the model for 600k steps without GAN loss, setting $\lambda_{lpips} = 4, \lambda_{adv} = 0, \lambda_{KL} = 1e-7$. The learning rate is warmed up to 5e-5 and decayed to 1e-5 using a cosine scheduler. After that, We continue training for another 100k steps with a learning rate of 1e-5 for the VAE and the discriminator, setting $\lambda_{lpips} = 4, \lambda_{adv} = 0.2, \lambda_{KL} = 1e-7$. We employ the Adam [17] optimizer ($\beta_1 = 0.5$ and $\beta_2 = 0.9$) and train the model with a batch size of 40 on 8 × NVIDIA A40 GPUs.

Evaluation Datasets and Metrics Following [1], we evaluate the reconstruction performance of the video VAEs on two video benchmarks: DAVIS [25] and TokenBench [1]. We employ four evaluation metrics: Structural Similarity Index Measure (SSIM) [35], Peak Signal-to-Noise Ratio (PSNR) [14], Learned Perceptual Image Patch Similarity (LPIPS) [46], and reconstruction Frechet Video Distance (rFVD) [31]. To assess performance across different video resolutions, we resize the videos in the evaluation set.

Method	Param.	Chn.	DAVIS				TokenBench			
			PSNR (↑)	**SSIM**(↑)	**LPIPS** (↓)	**rFVD** (↓)	**PSNR** (↑)	**SSIM** (↑)	**LPIPS**(↓)	**rFVD** (↓)
CV-VAE	182M	4	25.75	0.7345	0.1464	598.55	30.37	0.8758	0.0706	265.99
Open-Sora VAE	393M	4	**26.88**	**0.7741**	0.1582	611.11	<u>31.27</u>	0.8883	0.0774	282.81
OD-VAE	239M	4	26.16	0.7589	0.1173	407.20	30.47	0.8791	0.0618	211.93
VidTok	157M	4	<u>26.50</u>	<u>0.7707</u>	<u>0.1098</u>	<u>358.28</u>	**31.38**	**0.8963**	<u>0.0526</u>	<u>178.80</u>
LeanVAE (Ours)	40M	4	26.04	0.7629	**0.0899**	**322.46**	31.12	<u>0.8957</u>	**0.0432**	**162.55**
CogVideoX-VAE	206M	16	29.88	0.8741	0.0773	175.57	34.39	0.9366	0.0331	75.54
WF-VAE	316M	16	29.62	0.8668	0.0628	149.27	35.11	0.9435	0.0222	48.03
Cosmos-Tokenizer	101M	16	29.09	0.8477	0.1412	241.78	33.13	0.9191	0.0669	109.27
VidTok	157M	16	**31.06**	**0.8944**	**0.0436**	**103.79**	**36.12**	**0.9511**	**0.0166**	**40.89**
LeanVAE (Ours)	40M	16	<u>30.15</u>	<u>0.8760</u>	<u>0.0461</u>	<u>119.48</u>	<u>35.71</u>	<u>0.9506</u>	<u>0.0173</u>	<u>44.63</u>

Table 1. **Quantitative comparison on video reconstruction performance.** Chn indicates the dimension of latent channels. All evaluated models are causal and have a video compression ratio of $4 \times 8 \times 8$. The input resolution in DAVIS is $17 \times 256 \times 256$, with the encoded latent of the size $5 \times 32 \times 32$. The input video in TokenBench are resized with the short size of 256. The highest result is highlighted in **bold**, and the second highest result is <u>underlined</u>.

Videos in the DAVIS dataset are processed into a square resolution, while videos in TokenBench are resized based on the shorter side, preserving the original aspect ratio.

To assess our models on video generation, we train the video diffusion model following Latte [23] for 256×256 video generation with 16 frames.[2] We choose Latte-XL as the denoiser and replace its original VAE (SD VAE) with LeanVAE. The UCF101 [29] and SkyTimelapse [39] are used for class-conditional and unconditional video generation respectively. Following the evaluation guidelines in previous works [23], we use Frechet Video Distance (FVD) [31] to measure the quality of generated videos, computing FVD scores based on 2,048 video clips.

5.2. Video Reconstruction

Reconstruction Performance To demonstrate the effectiveness of LeanVAE in video reconstruction, we compare it with several state-of-the-art video autoencoders. The baseline models include CV-VAE [47], Open-Sora VAE [48], WF-VAE [22], CogVideoX VAE [42], VidTok [30], and Cosmos Tokenizer [1]. VidTok and Cosmos Tokenizer offer both continuous and discrete models; we evaluate only their continuous versions. For all models, we evaluate their latest open-source updates to ensure optimal performance.

The results are presented in Tab. 1. LeanVAE achieves highly competitive reconstruction performance with a significantly lean model size. For latent dimensions of 4, our model outperforms all baselines in LPIPS and rFVD metrics, demonstrating superior video quality. It also maintains competing performance in video fidelity, achieving comparable PSNR and SSIM scores with most baselines. When the latent channel is increased to 16, LeanVAE surpasses most models in reconstruction metrics but lags slightly behind VidTok. However, we argue that at 16 latent channels, both LeanVAE and VidTok exhibit strong reconstruction capabilities, and the visual quality gap between them is relatively negligible. We also provide a qualitative comparison of reconstruction results with competitive baselines in Fig. 2. As can be seen, our model excels in preserving fine details and capturing motion dynamics. These results underscore the effectiveness of LeanVAE in video reconstruction tasks.

Multi-resolution Evaluation We further compare LeanVAE with the best-performing baseline, VidTok, across multiple video resolutions. The evaluation focuses on computational cost and reconstruction performance, with results shown in Fig. 3. As shown in Fig. 3(a), LeanVAE demonstrates remarkable computational efficiency, achieving 50× fewer FLOPs than VidTok and accelerating inference speeds by 8-44× across resolutions. VidTok requires 94.1 TFLOPs for processing a 768^2 video with 17 frames, demanding impractical GPU memory. Given these, VidTok and similar models rely on tiling inference[3], further slowing down the speed. For instance, VidTok takes 20.26 seconds to process 17 frames of 768^2 video using 5-frame temporal tiling inference, whereas LeanVAE completes in just 0.46 seconds without chunking the frames.

In addition to its superior inference efficiency, LeanVAE maintains satisfying reconstruction quality, as evidenced in Fig. 3(b). At 768^2 resolution, it achieves near-identical PSNR ($\Delta < 0.3$dB) to VidTok while delivering better rFVD (18.6 and 6.3 points lower for 4 and 16 channels, respectively). It is also worth noting that we observed tiling in-

[2]We actually train the diffusion model on 17 frames and discard the first frame of the generated video to ensure consistency with the baselines during evaluation.

[3]This involves partitioning the video into chunk groups along temporal [1, 8, 22, 30] or spatial dimension [42, 47], followed by sequential inference on each chunk. Caching or overlapping group strategies are employed to maintain convolution continuity, see Appendix E.

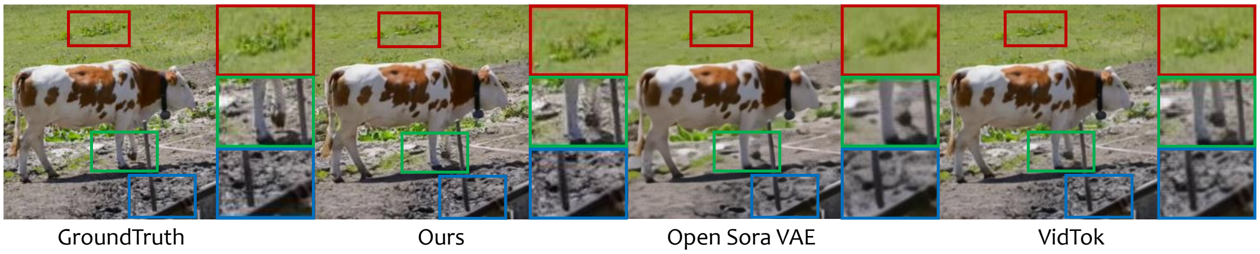

Figure 2. **Qualitative comparison between LeanVAE and leading baselines.** Due to space limitations, we present only the model with a latent channel size 4. The reconstruction performance of the leading models with 16 latent channels is notably better, and their visual differences are subtle. More comprehensive visual comparisons are available on our GitHub page.

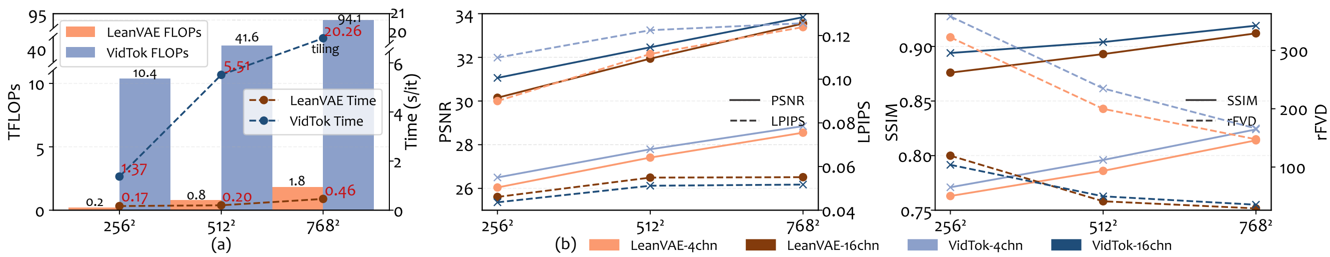

Figure 3. **Comparison across multiple resolutions**. (a) Computational cost in terms of TFLOPs(bar plots, labeled in black) and encoding-decoding time (line plots, labeled in red). (b) Reconstruction quality metrics. All evaluations were conducted on 17-frame videos using a single NVIDIA A40 (48GB) GPU.

ference may affect the reconstruction quality of VidTok. A well-designed caching strategy can enable lossless temporal titling inference [22], and LeanVAE's structure design supports a simple and efficient temporal caching solution. Further implementation details are provided in Appendix E.

Efficiency Analysis We conduct a comprehensive evaluation of inference efficiency across all these Video VAEs, reporting FLOPs, inference time, and memory consumption. The results are summarized in Tab. 4 (see Appendix B), offering compelling support for LeanVAE's superior computational performance over existing alternatives.

Overall, our rigorous evaluation demonstrates that LeanVAE achieves unparalleled efficiency in video reconstruction while maintaining competitive reconstruction quality.

5.3. Video Generation Evaluation

We evaluate video generation performance in Tab. 2 and present generated examples in Fig. 5. As shown, Latte equipped with LeanVAE (using 4 latent channels) achieves a further improvement in generation quality, attaining the best FVD scores for both conditional and unconditional generation. Additionally, due to the 4× temporal compression of inputs and the extra-efficient video encoding, LeanVAE supports a 4× larger batch size and achieves 6.64 samples/sec throughput, leading in a 315% increase in training speed over the original Latte.

We also notice that increasing the latent channels of LeanVAE to 16, while significantly improving video reconstruction, does not yield corresponding gains in generation performance. The FVD scores actually increased by 45.56 and 10.88, and we observed more distortion in the generated video. This finding aligns with work [43] that emphasizes the necessity for stricter regularization in high-channel latent spaces, indicating that high-channel diffusion remains an open research area and warrants further investigation.

Method	Sample/Sec	SkyTimelapse	UCF101
VideoGPT [41]	-	222.7	2880.6
StyleGAN-V [28]	-	79.52	1431.0
LVDM [13]	-	95.20	372.0
Latte [23] (4 chn)	1.60	59.82	477.97
Ours-Latte (4 chn)	6.64	**49.59**	**164.45**
Ours-Latte (16 chn)	6.64	95.15	175.33

Table 2. **FVD values of video generation on UCF101 and SkyTimelapse.** FVD for other baselines are referenced from Latte [23] paper. Sample/Sec is measured at the maximum batch size (2 for original Latte and 8 for our Latte) on one A40 (48GB).

5.4. Ablation Study

We conduct comprehensive ablation experiments to validate the superiority of the proposed techniques. In all experiments, we set the latent channel dimension to 4 and trained the model for 250k steps. The evaluation is conducted on DAVIS, and the results are summarized in Tab. 3.

Model Architecture We aim to determine the optimal *Encoder* and *Decoder* configurations for better model performance. To this end, we tested three architectures, including: **Variant 1**. Processes low-frequency and high-frequency components jointly; **Variant 2**. Process low-frequency and high-frequency components separately, then merge them; **Variant 3**. No wavelet transform was applied.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	rFVD ↓
(a) Variant 1	26.04	0.755	0.143	475.17
Variant 2	26.18	0.756	0.145	470.64
Variant 3	25.41	0.736	0.133	453.82
(b) AE	25.79	0.744	0.163	535.18
(c) w/ norm	22.91	0.703	0.158	599.38

Table 3. **Ablation study on different components**. Variant 2 (highlighted in gray) serves as the baseline across all groups, with CS channel compression and without patch normalization.

Figure 4. **Examples of block artifacts in reconstructed video**. Left:ground truth; Middle:reconstruction of model w/o pixel normalization; Right:reconstruction of model w/ pixel normalization.

Figure 5. **Examples of videos generated by Latte+LeanVAE (chn 4).** The top row is from SkyTimelapse and the bottom is UCF101. More examples are provided on the GitHub page.

The model architectures are plotted in Appendix F. All variants were designed to have similar parameter numbers to ensure a fair comparison. As shown in Tab. 3(a), Variant 2 performs the best fidelity with highest PSNR and SSIM, despite yielding slightly inferior LPIPS and rFVD scores. Given that these perceptual metrics can be greatly optimized through subsequent adversarial training, we choose Variant 2 as the final architecture. The following ablation studies were also conducted based on Variant 2.

AutoEncoding vs. Compressed Sensing We explore the advantages of using compressed sensing (CS) to construct the latent channel bottleneck. Therefore, we implement the bottleneck using both AE and CS algorithms. The two approaches have nearly the same parameter counts, except for the additional two scalars $\rho^{(k)}$ and $\theta^{(k)}$ in each iteration of the CS method. As shown in Tab. 3(b), the CS algorithm *ISTA-Net*$^{+}$ demonstrates significant performance gains, outperforming the AE-based approach by 0.39 dB in PSNR and 0.018 in LPIPS. These results highlight the potential of compressed sensing in this application.

Effect of Patch Normalization Block artifacts manifest as grid-like discontinuities in the visual output. We identified the LayerNorm commonly applied to pixel patches in transformers contributes to this issue. We evaluate the model with and without the patch normalization in Tab. 3(c), showing that normalization leads to a significant decline (3.27 dB in PSNR and 0.013 in LPIPS) in reconstruction metrics. Fig. 4 visualizes the block artifacts induced by pixel normalization. We hypothesize normalizing pixel patches introduces anisotropy, making it more challenging for the model to maintain consistency across reconstructed patches. We hope this finding could provide valuable insights for related models in low-level vision tasks.

6. Conclusion

We introduced **LeanVAE**, a novel and highly efficient Video VAE framework designed to address computational bottlenecks of video compression in Latent Video Diffusion Models. LeanVAE strikes an impressive balance between efficiency and reconstruction quality through the following contributions: (1) a lightweight architecture based on non-overlapping patching and a Neighborhood-Aware Feedforward (NAF) module, (2) the integration of wavelet transforms to enrich input representations, (3) the pioneering application of compressed sensing for channel compression, and (4) an analysis of the model design to identify the optimal architecture. Experiments have verified the boost LeanVAE brings to both video reconstruction and generation.

Limitations Future work includes developing models with higher compression rates, exploring discrete latent space representations, and exploring broader applications of LeanVAE in LVDMs, such as text-to-video generation. These efforts aim to further enhance LeanVAE's capacity and broaden its applicability in the visual generation field.

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. 1, 2, 3, 5, 6

[2] Dawit Mureja Argaw, Xian Liu, Qinsheng Zhang, Joon Son Chung, Ming-Yu Liu, and Fitsum Reda. High-quality joint image and video tokenization with causal vae. In *The Thirteenth International Conference on Learning Representations*. 2

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 2

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1

[6] C Sidney Burrus, Ramesh A Gopinath, and Haitao Guo. Wavelets and wavelet transforms. *rice university, houston edition*, 98, 1998. 2

[7] 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

[8] Liuhan Chen, Zongjian Li, Bin Lin, Bin Zhu, Qian Wang, Shenghai Yuan, Xing Zhou, Xinghua Cheng, and Li Yuan. Od-vae: An omni-dimensional video compressor for improving latent video diffusion model. *arXiv preprint arXiv:2409.01199*, 2024. 1, 2, 6

[9] M.C.Q. Farias and S.K. Mitra. No-reference video quality metric based on artifact measurements. In *IEEE International Conference on Image Processing 2005*, pages III–141, 2005. 1

[10] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, pages 102–118. Springer, 2022. 2

[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. 3

[12] Philippe Hansen-Estruch, David Yan, Ching-Yao Chung, Orr Zohar, Jialiang Wang, Tingbo Hou, Tao Xu, Sriram Vishwanath, Peter Vajda, and Xinlei Chen. Learnings from scaling visual tokenizers for reconstruction and generation. *arXiv preprint arXiv:2501.09755*, 2025. 1, 3

[13] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 7

[14] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, 2010. 5

[15] Jin-Young Jeong, Mustafa Ozger, and Woong-Hee Lee. Compressed sensing verses auto-encoder: On the perspective of signal compression and restoration. *IEEE Access*, 12: 41967–41979, 2024. 2

[16] Kihong Kim, Haneol Lee, Jihye Park, Seyeon Kim, Kwanghee Lee, Seungryong Kim, and Jaejun Yoo. Hybrid video diffusion models with 2d triplane and 3d wavelet representation. In *European Conference on Computer Vision*, pages 148–165. Springer, 2024. 2, 3, 4

[17] DiederikP. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv: Learning,arXiv: Learning*, 2014. 5

[18] DiederikP. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv: Machine Learning,arXiv: Machine Learning*, 2013. 5

[19] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, 2024. 1, 2

[20] Binglin Li, Jie Liang, and Jingning Han. Variable-rate deep image compression with vision transformers. *IEEE Access*, 10:50323–50334, 2022. 2, 4

[21] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *ArXiv*, abs/2406.11838, 2024. 5

[22] Zongjian Li, Bin Lin, Yang Ye, Liuhan Chen, Xinhua Cheng, Shenghai Yuan, and Li Yuan. Wf-vae: Enhancing video vae by wavelet-driven energy flow for latent video diffusion model. *arXiv preprint arXiv:2411.17459*, 2024. 1, 2, 3, 6, 7

[23] 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. 1, 6, 7

[24] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li,

Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2024. 1, 2

[25] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 5

[26] 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. 1

[27] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. Litevae: Lightweight and efficient variational autoencoders for latent diffusion models. *arXiv preprint arXiv:2405.14477*, 2024. 2, 3

[28] 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. 7

[29] Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv: Computer Vision and Pattern Recognition,arXiv: Computer Vision and Pattern Recognition*, 2012. 6

[30] Anni Tang, Tianyu He, Junliang Guo, Xinle Cheng, Li Song, and Jiang Bian. Vidtok: A versatile and open-source video tokenizer. *arXiv preprint arXiv:2412.13061*, 2024. 1, 3, 6, 2

[31] Thomas Unterthiner, Sjoerdvan Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. *International Conference on Learning Representations,International Conference on Learning Representations*, 2019. 5, 6

[32] 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 descriptions. In *International Conference on Learning Representations*, 2022. 2

[33] Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *arXiv preprint arXiv:2406.09399*, 2024. 1, 3

[34] Yuchi Wang, Junliang Guo, Xinyi Xie, Tianyu He, Xu Sun, and Jiang Bian. Vidtwin: Video vae with decoupled structure and dynamics. *arXiv preprint arXiv:2412.17726*, 2024. 3

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[36] Zijian Zhang Rox Min Zuozhuo Dai Jin Zhou Jiangfeng Xiong Xin Li Bo Wu Jianwei Zhang Kathrina Wu Qin Lin Aladdin Wang Andong Wang Changlin Li Duojun Huang Fang Yang Hao Tan Hongmei Wang Jacob Song Jiawang Bai Jianbing Wu Jinbao Xue Joey Wang Junkun Yuan Kai Wang Mengyang Liu Pengyu Li Shuai Li Weiyan Wang Wenqing Yu Xinchi Deng Yang Li Yanxin Long Yi Chen Yutao Cui Yuanbo Peng Zhentao Yu Zhiyu He Zhiyong Xu Zixiang Zhou Zunnan Xu Yangyu Tao Qinglin Lu Songtao Liu Dax Zhou Hongfa Wang Yong Yang Di Wang Yuhong Liu Weijie Kong, Qi Tian and along with Caesar Zhong Jie Jiang. Hunyuanvideo: A systematic framework for large video generative models, 2024. 1

[37] Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video vae for latent video diffusion model. *arXiv preprint arXiv:2411.06449*, 2024. 2

[38] Yazhou Xing, Yang Fei, Yingqing He, Jingye Chen, Jiaxin Xie, Xiaowei Chi, and Qifeng Chen. Large motion video autoencoding with cross-modal video vae. *arXiv preprint arXiv:2412.17805*, 2024. 2

[39] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. *Cornell University - arXiv,Cornell University - arXiv*, 2017. 6

[40] Jiaqi Xu, Xinyi Zou, Kunzhe Huang, Yunkuo Chen, Bo Liu, MengLi Cheng, Xing Shi, and Jun Huang. Easyanimate: A high-performance long video generation method based on transformer architecture. *arXiv preprint arXiv:2405.18991*, 2024. 1, 2

[41] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 7

[42] 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. 1, 2, 6

[43] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025. 7

[44] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024. 2, 5

[45] Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018. 4, 5

[46] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 5

[47] Sijie Zhao, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Muyao Niu, Xiaoyu Li, Wenbo Hu, and Ying Shan. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*, 2024. 1, 2, 6

[48] 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, 2024. 1, 2, 6

[49] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 5