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

Appendix

A. Overview

In this appendix, we provide additional details and explanations as follows:

- We present a comprehensive comparison of the computational efficiency of Video VAEs across two resolutions (256^2 and 512^2). We report FLOPs, inference time, and memory consumption for each, further validating LeanVAE's superior efficiency. Results are summarized in Tab. 4.
- We visualize the transformation of videos into the frequency domain using the Haar Discrete Wavelet Transform (DWT) and the subsequent reconstruction via the Inverse Discrete Wavelet Transform (IDWT) in Fig. 6.
- A straightforward comparison between AutoEncoding (AE) and Compressed Sensing (CS) signal recovery algorithms is presented in Appendix D.
- We introduce the tiling inference technique utilized in Video VAEs in Appendix E, along with its influence on VidTok's reconstruction results in Tab. 5. Furthermore, we elaborate on the caching strategy adopted by LeanVAE that supports lossless temporal tiling inference.
- The architectural details of three LeanVAE variants are plotted in Fig. 9.

B. Comprehensive Efficiency Comparison

	CV-VAE	Open-Sora	OD-VAE	VidTok	CogVideoX	WF-VAE	Cosmos	LeanVAE
TFLOPs	11.4 / 45.8	9.9 / 40.2	14.4 / 57.9	10.4 / 41.6	15.5 / 61.9	6.9 / 27.8	0.6 / 2.4	0.2 / 0.8
Time (s/it)	0.76 / 3.25	0.80 / 3.09	0.91 / 3.84	1.37 / 5.51	1.29 / 5.05	0.52 / 2.69	0.08-0.13 / 0.37	0.04-0.17 / 0.20
Mem. (GiB)	13.1 / 26.8	2.5 / 4.0	13.4 / 37.4	12.8 / 34.6	9.6 / 34.6	4.2 / 13.3	2.0 / 3.0	2.0 / 2.9

Table 4. Efficiency comparison across all models on 256^2 / 512^2 resolution with 17 frames. An anomalously slower first-round evaluation was observed only for Cosmos and LeanVAE at 256px. We thus report a range, with the upper bound reflecting the warm-up round and the lower representing subsequent stable rounds.

It is well known that architecture design is challenging, especially given the high training cost of Video VAEs. We break two dominant constraints in existing design paradigms: (1) the dependence on SD VAE-style spatial priors, and (2) the reliance on global attention computation. Our work shows that neither constraint is essential for video reconstruction: the former introduces significant parameter and computation redundancy, the latter adds computation overhead. By discarding both, LeanVAE achieves notable gains in efficiency, as shown in Tab. 4. While Cosmos also shows notable efficiency, its reconstruction lags far behind peer baselines, likely suggesting our idea is more effective than its simplification strategy.

C. Visualization of Haar Wavelet Transform

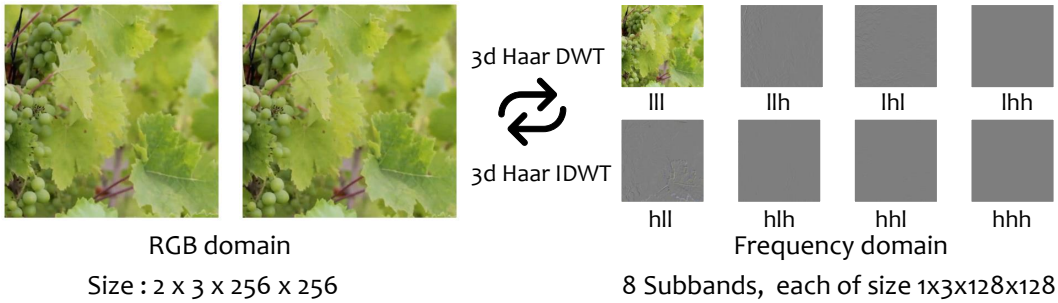

Figure 6. Visualization of the 3D Haar wavelet transform.

D. Comparison between AE and CS Signal Recovery Algorithms

AutoEncoding (AE) and Compressed Sensing (CS) are two classical paradigms for data compression and reconstruction. In AE algorithms, data is compressed through an encoder network and subsequently restored by a decoder network. CS methods perform downsampling by multiplying data with a sensing matrix, the signal is then reconstructed by a recovery algorithm. A comparison of these two frameworks is illustrated in Fig. 7.

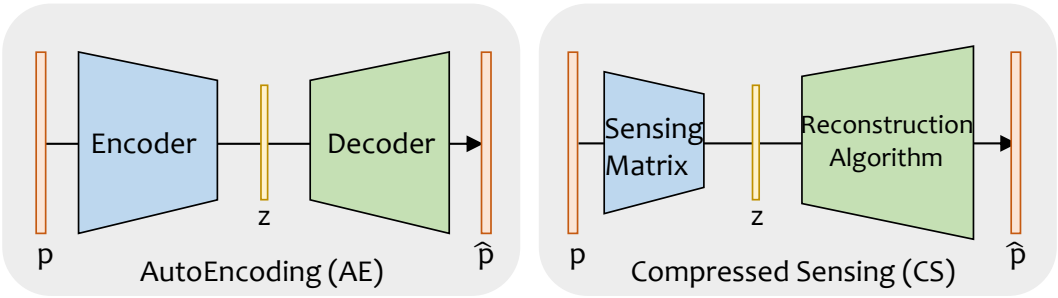

Figure 7. Comparison of AE and CS frameworks.

E. Tiling Inference

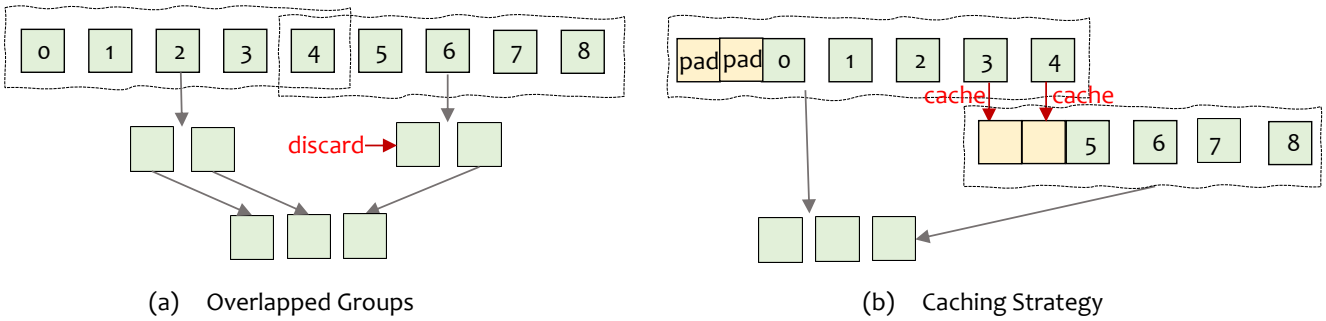

Figure 8. Two approaches to mitigating discontinuities in temporal tiling inference. Only temporal changes are shown here. (a) Overlapped group processing, (b) Caching mechanism in convolution modules. The illustration shows chunksize=5, overlap count=1, and cache count=2.

Tiling inference is a widely adopted technique to enable inference on long video sequences or high-resolution frames when constrained by memory limitations. The core idea of tiling inference is to partition the video into smaller chunks, along the temporal [1, 8, 22, 30] or spatial dimension [42, 47], with inferenced results subsequently concatenated. However, tiling inference inevitably introduces discontinuities at chunk boundaries due to the lack of contextual information from adjacent chunks during convolution. This discontinuity can lead to visible artifacts or temporal inconsistency in the reconstructed video.

In this work, we focus on temporal tiling inference, which can mitigate such discontinuities using the properties of causal CNNs. Two common methods are employed to address this issue: (1) Overlapped Groups: As shown in Fig. 8(a), frames are grouped with some overlap between adjacent chunks. (2) Caching Strategy: In convolution, previous inference results are cached as padding for the next chunk, as illustrated in Fig. 8(b). The effectiveness of these approaches depends critically on the chunksize and overlap/cache counts, which influence both the computational efficiency and reconstruction results.

In our multi-resolution experiments (Sec. 5.2), as VidTok does not offer caching support, we employ overlapped group methods. At 768^2 resolution, the maximum feasible chunksize for VidTok is 5 and we set overlap counts to 1. Despite using the overlapped groups method, we observe that VidTok's reconstruction is still impacted by tiling inference, as shown in Tab. 5. With smaller chunksize , VidTok's memory usage decreases, but the number of inference steps increases accordingly.

Recent work [22] introduced a model-aware caching strategy for lossless temporal tiling inference. For the LeanVAE model, which consists only of causal depthwise convolutions with a kernel size of 3x3x3 and stride 1x1x1, the lossless caching strategy is straightforward: the last two tokens in previous chunk need to be cached in each convolution structure

(the first chunk still uses zero padding). The detailed computation process can be found in [22]. This approach maintains consistent reconstruction quality across varying chunksizes (Tab. 5). Furthermore, as LeanVAE's memory usage does not vary significantly between 5-frame and 17-frame video inference, the caching operation actually increases memory consumption than full inference.

Discussion on Caching Precomputed Latents. An alternative to tiling inference is to precompute and cache all video latents before diffusion model training. This eliminates the repeated use of the VAE encoder during training. However, such caching brings several drawbacks [27]: it incurs substantial storage overhead, disables on-the-fly data augmentation, and limits the flexibility of frame sampling strategies for training samples. Hence, an efficient encoder like LeanVAE remains indispensable for scalable and flexible diffusion training.

Model	PSNR ↑	SSIM ↑	LPIPS ↓	rFVD ↓	Mem.(GB)
VidTok (chunksize=5)	26.53	0.7708	0.1086	342.44	6.5
VidTok (chunksize=9)	26.48	0.7692	0.1101	351.44	8.9
VidTok (Full)	26.50	0.7707	0.1098	358.28	13.8
LeanVAE (chunksize=5)	26.04	0.7629	0.0899	322.56	2.1
LeanVAE (chunksize=9)	26.04	0.7629	0.0900	322.58	2.3
LeanVAE (Full)	26.04	0.7629	0.0899	322.46	2.3

Table 5. Performance with temporal tiling inference. Evaluation is conducted on 17-frame 256×256 DAVIS videos. "Full" denotes inference without tiling.

F. Architectural Variants

We explored three architectural variants of LeanVAE.The detailed network architectures are illustrated in Fig. 9.

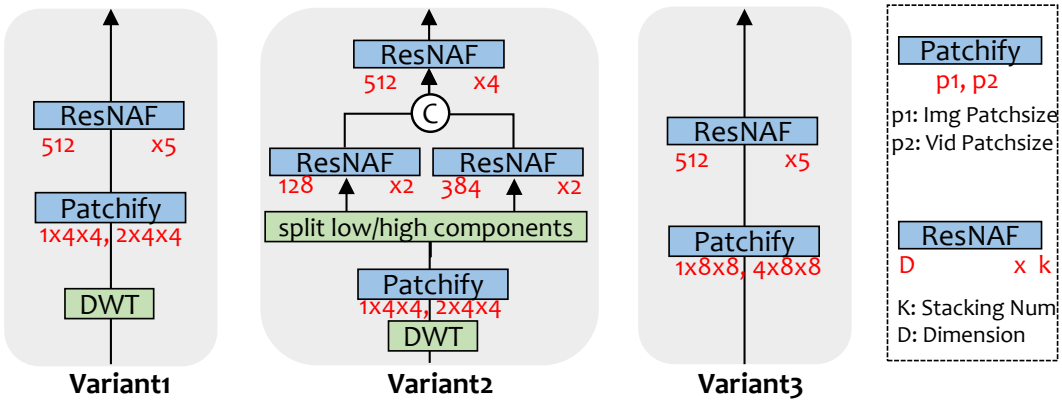

Figure 9. Architectural overview of LeanVAE variants (Patching and *Encoder*). The Unpatching and *Decoder* adopt symmetric structures.

- **Variant 1**: Jointly processes low-frequency and high-frequency components.
- **Variant 2**: Separately processes low-frequency and high-frequency components, followed by merging them.
- **Variant 3**: Directly processes the input without wavelet decomposition.