

Turbo-VAED: Fast and Stable Transfer of Video-VAEs to Mobile Devices

Ya Zou*, Jingfeng Yao*, Siyuan Yu, Shuai Zhang, Wenyu Liu, Xinggang Wang[†]

Huazhong University of Science and Technology

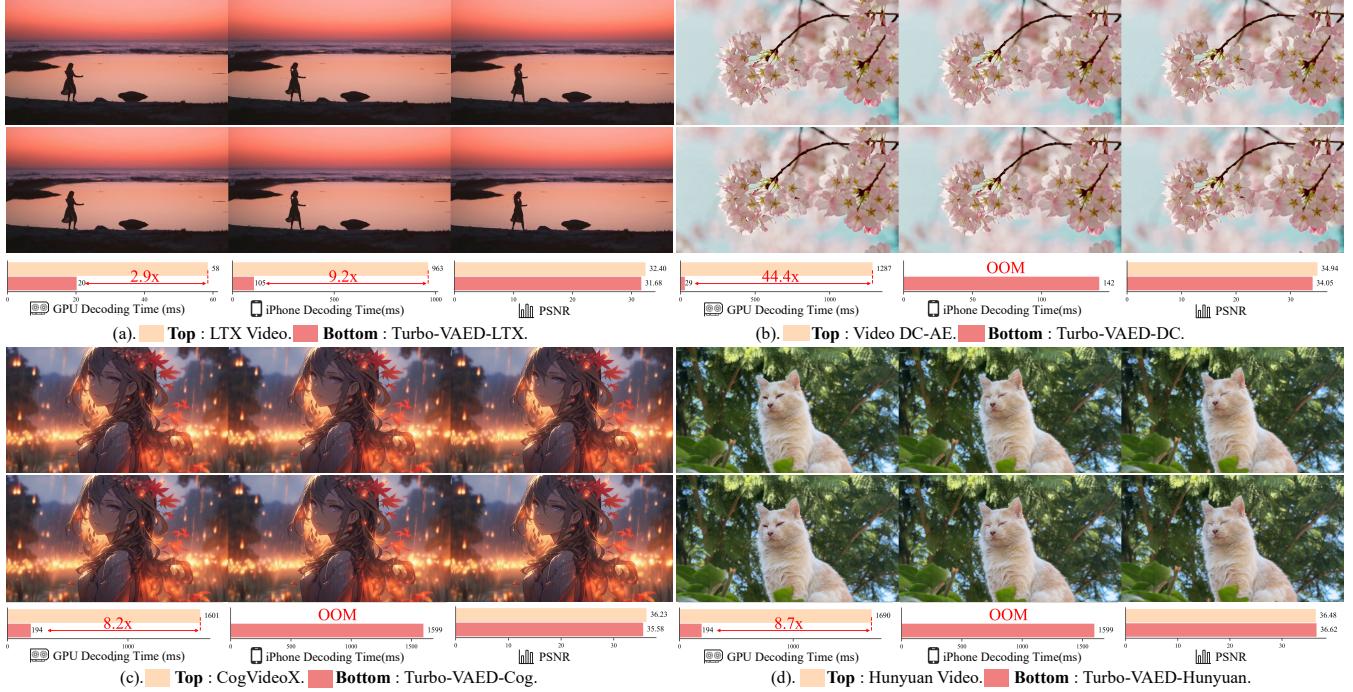


Figure 1: Video Reconstruction Results. We compare four widely used video VAEs and their “turbo-charged” variants. The upper section displays reconstructed videos (Top: base model; Bottom: Turbo-VAED version). The turbo-charged decoders significantly reduce decoding latency across both GPU and iPhone platforms while maintaining reconstruction quality.

Abstract

There is a growing demand for deploying large generative AI models on mobile devices. For recent popular video generative models, however, the Variational AutoEncoder (VAE) represents one of the major computational bottlenecks. Both large parameter sizes and mismatched kernels cause out-of-memory errors or extremely slow inference on mobile devices. To address this, we propose a low-cost solution that efficiently transfers widely used video VAEs to mobile devices. (1) We analyze redundancy in existing VAE architectures and get empirical design insights. By integrating 3D depthwise separable convolutions into our model, we significantly reduce the number of parameters. (2) We observe that the upsampling techniques in mainstream video VAEs are poorly suited to mobile hardware and form the main bottle-

neck. In response, we propose a decoupled 3D pixel shuffle scheme that slashes end-to-end delay. Building upon these, we develop a universal mobile-oriented VAE decoder, **Turbo-VAED**. (3) We propose an efficient VAE decoder training method. Since only the decoder is used during deployment, we distill it to Turbo-VAED instead of re-training the full VAE, enabling fast mobile adaptation with minimal performance loss. To our knowledge, our method enables *real-time 720p video VAE decoding on mobile devices for the first time*. This approach is widely applicable to most video VAEs. When integrated into four representative models, with training cost as low as \$95, it accelerates original VAEs by up to 84.5 \times at 720p resolution on GPUs, uses as low as 17.5% of original parameter count, and retains 96.9% of the original reconstruction quality. Compared to mobile-optimized VAEs, Turbo-VAED achieves a 2.9 \times speedup in FPS and better reconstruction quality on the iPhone 16 Pro. The code and models will soon be available at <https://github.com/hustvl/Turbo-VAED>.

*Equal contribution.

[†]Corresponding author: xgwang@hust.edu.cn

1 Introduction

Driven by the growing demand for deploying large generative AI models on mobile devices (Marafioti et al. 2025; Hu et al. 2024; Team et al. 2023), adapting video generation models for mobile platforms has attracted considerable attention (Wu et al. 2025b; Yahia et al. 2024). As a key component in latent diffusion models (Rombach et al. 2022), VAEs (Kingma, Welling et al. 2013) compress visual signals into latent spaces. However, most current video VAEs are incompatible with mobile devices.

The pursuit of better visual compression capability has driven VAEs to scale up. For instance, LTX-VAE (HaCohen et al. 2024) and Video DC-AE (Peng et al. 2025) reach over four times the size of SVD-VAE (Blattmann et al. 2023). Large model sizes always cause out-of-memory (OOM) errors on mobile devices. Additionally, incompatible operators result in unacceptably slow inference. The 3D pixel shuffle module is widely adopted in video VAEs (HaCohen et al. 2024; Wu et al. 2025a) for upsampling. However, it suffers from poor mobile compatibility, exhibiting a standalone latency that is $11\times$ greater than that of our mobile-optimized operator. Consequently, lightweight models incorporating mobile-optimized operations are required to enable real-time inference.

While training a lightweight VAE from scratch is a potential solution, it demands substantial computational resources. Moreover, compact models learn latent distributions that are markedly inferior to those of larger models. To address this, the decoder-only distillation method (Wu et al. 2025b) provides a viable direction through initial research, with room for further in-depth analysis.

In this paper, we propose **Turbo-VAED**, a family of lightweight VAE decoders optimized for mobile deployment. Its architecture effectively reduces model redundancy and parameter count, while our mobile-friendly upsampling strategy substantially reduces on-device inference latency. Our comprehensive experiments and analysis of the decoder-only distillation method, while methodologically straightforward, yield key empirical insights enabling efficient and generalizable transfer of video VAEs to mobile devices. Specifically, we conduct the following work:

Mobile Model Design (Sec. 3.2 3.3) We design a universal mobile VAE decoder incorporating the following key insights. (1) *Parameter-efficient Decoder*. Through experiments and analysis, we identify significant parameter redundancy in low-resolution layers of the VAE decoder. Integrating 3D depthwise separable convolutions into these layers substantially reduces model parameters while maintaining reconstruction quality. (2) *Mobile-friendly 3D Upsampling Strategy*. The two widely used 3D upsampling techniques are 3D pixel shuffle (high-quality but slow) and 3D interpolation (low-quality and unsupported on mobile devices). To accelerate execution speed while retaining the reconstruction quality as high as possible, we modify the 3D pixel shuffle by decoupling its spatial and temporal components.

Training Method (Sec. 3.4) Our training pipeline involves two main designs. (1) *Decoder-only Distillation*. Our approach involves freezing the pre-trained VAE encoder and

training a tiny decoder, preserving the high-quality latent representations unchanged. We adopt this strategy because text-to-video generation relies exclusively on the decoder to transform latents into videos. Furthermore, during diffusion model training, the encoder runs only once to convert the dataset into stored latents, while the decoder is executed repeatedly. (2) *High Data Efficiency and Negligible Cost via Feature Alignment*. We distill knowledge from the original decoder into the lightweight decoder by aligning its intermediate features. Our experiments show that training with this technique remains feasible even on limited datasets, requiring a cost as low as \$95.

Turbo-VAED Family (Sec. 4.1) To validate the broad generalizability of our model design and training method, we adopt Hunyuan-VAE (Kong et al. 2024), CogVideoX-VAE (Yang et al. 2024), Video DC-AE (Peng et al. 2025), and LTX-VAE (HaCohen et al. 2024) as teacher models. Their corresponding student models are named Turbo-VAED-Hunyuan, Turbo-VAED-Cog, Turbo-VAED-DC, and Turbo-VAED-LTX, respectively.

Evaluation (Sec. 4.3) We extensively evaluate Turbo-VAED. By reducing the parameter count to as low as 17.5% of the original model, the Turbo-VAED family achieves up to a $44.4\times$ speedup at 512px resolution and $84.5\times$ speedup at 720p resolution on the GPU. While achieving acceleration, they preserve up to 96.9% reconstruction performance and up to 97.3% generation performance. The lightweight design also enables the mobile deployment of previously incompatible large-scale models. Compared to mobile-optimized video VAEs like H3AE (Wu et al. 2025a), Turbo-VAED-DC achieves a $2.9\times$ speedup in FPS and better reconstruction quality under the same compression ratio on the iPhone 16 Pro. Notably, Turbo-VAED-DC and Turbo-VAED-LTX enable the *first* successful decoding of 720p videos on the iPhone at up to 38.1 FPS.

Our contributions are summarized as follows:

- We propose a universal mobile-oriented video VAE architecture design, featuring a parameter-efficient decoder and a mobile-friendly 3D upsampling strategy.
- We present an efficient distillation method for transferring video VAEs to mobile devices, with total training cost as low as \$95.
- We evaluate our method on four state-of-the-art video VAEs. The Turbo-VAED family reduces the parameter count to as low as 17.5% of the original model, achieving up to $84.5\times$ faster inference at 720p resolution on GPUs and maintaining up to 96.9% reconstruction performance. Our method enables the first real-time 720p video VAE decoding on the iPhone 16 Pro.

2 Related Work

2.1 Mobile Deployment of Large Models

The demand for deploying large models on mobile devices, such as large language models (LLMs) and diffusion models (Rombach et al. 2022; Peebles and Xie 2023; Yao et al.

2024), is increasing. For instance, LLMs (Hu et al. 2024; Liu et al. 2024; Team et al. 2023; Marafioti et al. 2025) achieve real-time on-device execution. (Wu et al. 2025b; Yahia et al. 2024; Kim et al. 2025) explore text-to-video generation for mobile devices. However, deploying video diffusion models on mobile platforms remains a challenge. A critical bottleneck lies in the VAE, which cause OOM errors or extremely slow inference. And retraining compact VAEs demands significant computational resources. To bridge this gap, we propose Turbo-VAED, a family of lightweight VAE decoders optimized for mobile deployment.

2.2 Video Autoencoders

Standard autoencoder (Bank, Koenigstein, and Giryes 2023) learns latent representations for reconstruction, while VAE introduces probabilistic modeling via latent distribution constraints. VQ-VAE (Van Den Oord, Vinyals et al. 2017) employs codebook-based representation discretization, and VQGAN (Esser, Rombach, and Ommer 2021) integrates adversarial training (Goodfellow et al. 2020). These autoencoders (Chen et al. 2024; Yao, Yang, and Wang 2025) underpin modern diffusion models by compressing pixel data into latents for efficient denoising. Notably, the community has proposed numerous high-performance video VAEs. Some models (Yu et al. 2023) learn the distribution of discrete tokens. In contrast, most video VAEs model continuous latents. Early methods like (Blattmann et al. 2023) focus on spatial compression, while later works (Zheng et al. 2024; Polyak et al. 2024; Hansen-Estruch et al. 2025; Zhao et al. 2024; Xing et al. 2024; Tian et al. 2024) compress spatial and temporal dimensions for greater redundancy reduction. Recently, some models explore efficient inference (Cheng and Yuan 2025; Agarwal et al. 2025; Wu et al. 2025a). However, most high-quality models still fail to achieve real-time video decoding on mobile devices. We explore mobile-oriented model design and efficient transfer strategies, distilling these models into the Turbo-VAED family.

3 Method

In this section, we first propose our designs based on *parameter-efficient decoder* and *mobile-friendly 3D upsampling strategy*, which are universal for most video VAEs. Additionally, we introduce a fast distillation method and highlight its critical role during the training process.

3.1 Preliminary

Video VAEs To enable simultaneous compression of both videos and images into a unified latent space, most VAEs impose specific constraints on the number of input video frames. Given a video $X \in \mathbb{R}^{3 \times (T+1) \times H \times W}$, the VAE encodes it into a latent representation $L \in \mathbb{R}^{C \times (\frac{T}{d_t} + 1) \times \frac{H}{d_h} \times \frac{W}{d_w}}$, where d_t , d_h , and d_w denote the down-sampling factors for time, height, and width respectively.

3D Depthwise Separable Convolution Depthwise separable convolutions reduce computational cost and model size, enabling efficient deployment on resource-constrained devices (Howard et al. 2017; Sandler et al. 2018). The 3D

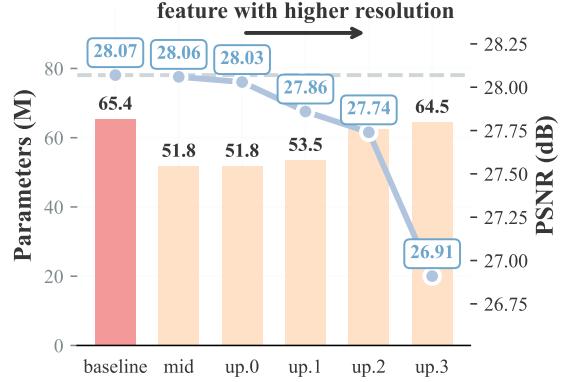


Figure 2: **Decoder Redundancy Analysis.** Experimental results demonstrate that lightweight modifications at higher feature resolutions yield less substantial parameter reduction and markedly degraded reconstruction performance.

depthwise separable convolution (3D DW Conv) is extended to 3D vision tasks and can be described as follows (Ye, Liu, and Zhang 2019):

$$\hat{\mathbf{G}}_{k,l,t,m} = \sum_{i,j,f} \hat{\mathbf{K}}_{i,j,f,m} \cdot \mathbf{F}_{k+i-1,l+j-1,t+f-1,m} \quad (1)$$

$$\mathbf{G}_{k,l,t,n} = \sum_{i,j,f,m} \mathbf{K}_{i,j,f,m,n} \cdot \hat{\mathbf{G}}_{k+i-1,l+j-1,t+f-1,m} \quad (2)$$

3.2 Reducing Parameter Redundancy

Decoder Redundancy Analysis To improve the parameter efficiency of the decoder network, we can replace naive 3D convolutions with depthwise separable convolutions across different layers. We start with a lightweight decoder composed entirely of standard 3D convolutions as the baseline to distill the LTX-VAE. We gradually apply the replacement from low-resolution, deep layers (e.g., *mid*, *up₀*) to high-resolution, top layers (e.g., *up₃*), and the results are shown in Figure 2. The experimental results show that applying lightweight modifications from low-resolution to high-resolution layers causes a gradual increase in parameter count toward the baseline, but the reconstruction quality progressively deteriorates, as indicated by the decreasing PSNR. This suggests that there are many redundant parameters in the low-resolution layers, but few in the high-resolution layers.

Finding 1: In the VAE decoder, network layers processing lower-resolution features exhibit higher parameter redundancy; employing depthwise separable convolutions in these layers significantly enhances parameter efficiency.

Parameter-efficient Decoder Our mobile decoder Turbo-VAED adopts a hybrid architecture, employing 3D depthwise separable convolutions in low-resolution layers and standard 3D convolutions in other layers. We perform replacements in *mid* and *up₀* layers, achieving a 41.6% reduction in parameters while maintaining virtually identical reconstruction performance (PSNR 28.05 vs. baseline 28.07).

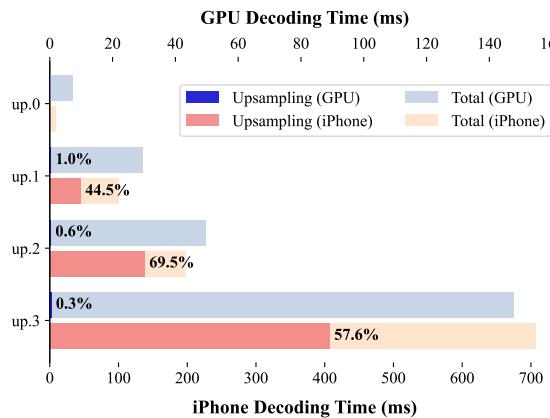


Figure 3: **Decoding Time in Different Blocks.** We conduct a thorough decoding time analysis per block. On mobile devices, the upsampling operation (3D pixel shuffle) incurs significant latency due to poor kernel compatibility, becoming the primary bottleneck in the decoding pipeline.

Upsampling	Decoding Time	PSNR↑	LPIPS↓	SSIM↑
3D Pixel Shuffle	1343 ms	28.05	0.1293	0.8431
3D Interpolate	/	27.40	0.1392	0.8272
Ours	446 ms	27.86	0.1312	0.8396

Table 1: **Upsampling Techniques.** We ablate different upsampling methods in the decoder architecture. Our approach achieves a balance between decoding speed and reconstruction quality.

3.3 Accelerating 3D Upsampling

Mobile Upsampling Latency Analysis The 3D pixel shuffle is widely used for upsampling in video VAEs (Ha-Cohen et al. 2024; Wu et al. 2025a). Given its ability to achieve superior reconstruction quality (Table 1), we initially incorporate it into our mobile decoder design. However, this model exhibits high inference latency on mobile devices. Therefore, we perform an in-depth decoding time analysis for each block, as shown in Figure 3. On GPUs, the execution time of 3D pixel shuffle accounts for a very small fraction of the decoding time per block. However, on mobile devices, it dominates the decoding time. This high-latency upsampling operation is the key factor that slows down the entire model’s on-device decoding speed.

Finding 2: The 3D Pixel Shuffle demonstrates low computational efficiency for upsampling on mobile devices due to poor kernel compatibility, emerging as the primary latency bottleneck during decoding.

Mobile-friendly 3D Upsampling Strategy Although 3D interpolation is a common alternative, it exhibits inferior reconstruction quality and lacks support in major mobile operator libraries. To achieve a decoder with fast inference speed, we propose a novel mobile-friendly upsampling solution.

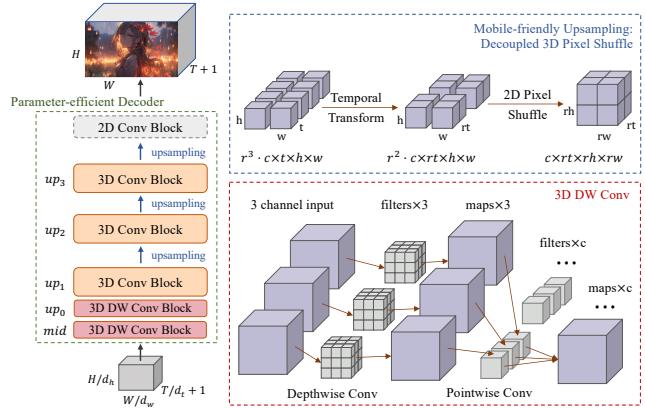


Figure 4: **Turbo-VAED Architecture Overview.** We illustrate the mobile-oriented architecture design: a parameter-efficient decoder that incorporates a mobile-friendly 3D upsampling strategy.

We decompose the 3D pixel shuffle into distinct temporal and spatial operations, as illustrated in the top-right of Figure 4. First, transform the convolution layers’ output $F \in \mathbb{R}^{(r^3 \times C) \times T \times H \times W}$ by converting channels to the temporal dimension, producing an intermediate feature $\hat{F} \in \mathbb{R}^{(r^2 \times C) \times rT \times H \times W}$, where r is the scaling factor. The spatial upsampling process involves applying 2D pixel shuffle (Shi et al. 2016), which can be formulated as follows to produce the final video $Y \in \mathbb{R}^{C \times rT \times rH \times rW}$:

$$Y_{c,t,h,w} = \hat{F}_{C \cdot r \cdot \text{mod}(w,r) + C \cdot \text{mod}(h,r) + c, t, \lfloor h/r \rfloor, \lfloor w/r \rfloor} \quad (3)$$

Our upsampling technique results in a significantly shortened execution chain of operators after compilation, leading to faster inference speed on mobile devices. As shown in Table 1, experiments validate that Turbo-VAED-LTX with our upsampling technique achieves a 66.8% speedup compared to its counterpart with 3D pixel shuffle on iPhone devices. While our method shows slightly inferior reconstruction quality compared to 3D pixel shuffle, it outperforms 3D interpolation. Therefore, we adopt this mobile-friendly design as the 3D upsampling strategy in Turbo-VAED.

3.4 Enhancing Training Efficiency

Distillation Loss Analysis To obtain an efficient training method that transfers the pre-trained video VAEs to mobile devices, we employ knowledge distillation from the original decoder to Turbo-VAED. Following prior knowledge distillation works (Yang et al. 2022b; Bai et al. 2023; Yang et al. 2022a; Doshi and Kim 2024; Touvron et al. 2021), we design a distillation loss that aims to align the intermediate layer features of the two decoders, as defined in Equation 4.

$$L_{\text{distill}} = \sum_l \frac{1}{\text{numel}(f_l^T)} \sum_i \|\sigma(f_l^S)_i - f_{l,i}^T\|_1 \quad (4)$$

Where l denotes the number of blocks, $\text{numel}(\cdot)$ represents the total number of elements, f_l^T and f_l^S denote the features of the corresponding layers in the teacher and student

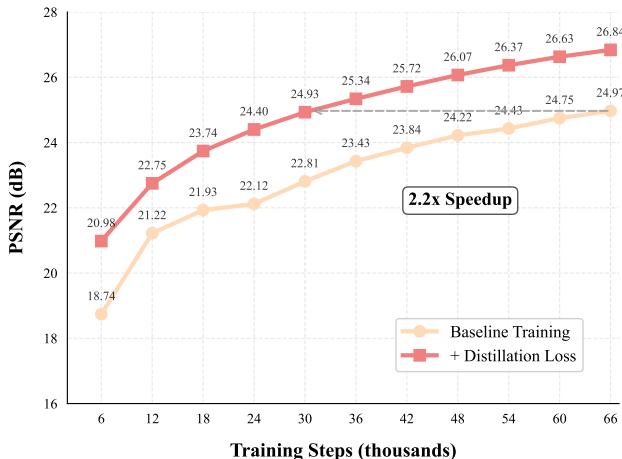


Figure 5: **Distillation Loss.** We train Turbo-VAED-LTX on VidGen dataset at 256px resolution, ablating the additional distillation loss. The distillation loss significantly accelerates convergence while enhancing reconstruction quality.

Dataset	Samples	PSNR↑	LPIPS↓	SSIM↑
Subset	10,000	29.21	0.0943	0.8709
Full	1,000,000	29.23	0.0950	0.8711

Table 2: **Number of Training Samples.** We investigate training with our distillation loss across varying dataset sizes. Performance with 10K and 1M samples is comparable, demonstrating our method’s low data requirements and high practical value.

decoders. And $\sigma(\cdot)$ refers to the projection network function, which maps student features to align with the teacher model’s hidden dimension.

As shown in Figure 5, incorporating $L_{distill}$ accelerates convergence with a 2.2x speedup. Training Turbo-VAED-LTX with $L_{distill}$ yields a PSNR of 30.39 at convergence on the VidGen test set (baseline: 28.77), demonstrating superior reconstruction quality. Furthermore, Table 2 highlights that models trained on 10k and 1M video datasets using our distillation loss achieve comparable performance.

Finding 3: Feature alignment-based distillation enables data-efficient training, substantially enhancing model performance while accelerating convergence.

Efficient Distillation Method As illustrated in Figure 6, we freeze the encoder and distill knowledge from the original decoder to Turbo-VAED by aligning intermediate layer features between them. In addition to standard reconstruction loss L_1 and KL loss L_{kl} , we incorporate the perceptual loss L_{lpips} , the adversarial GAN loss L_{adv} , and our designed distillation loss $L_{distill}$. The complete loss function is shown in Equation 5. Following the training strategy of (Peng et al. 2025), we employ a two-stage procedure: L_{adv} is excluded during the initial stage and introduced only after the model

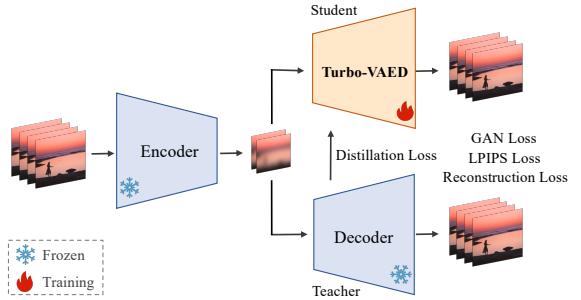


Figure 6: **Training Pipeline.** The pre-trained VAE remains frozen as we distill the lightweight decoder Turbo-VAED by aligning the intermediate features.

reaches near-convergence in the previous stage.

$$L = L_1 + \alpha_1 L_{lpips} + \alpha_2 L_{distill} + \alpha_3 L_{kl} + \alpha_4 L_{adv} \quad (5)$$

4 Experiments

4.1 Turbo-VAED Family

We employ SOTA video VAEs from (Kong et al. 2024; Yang et al. 2024; HaCohen et al. 2024; Peng et al. 2025) as teacher models for distillation. Hunyuan-VAE realizes near-lossless video fidelity. CogVideoX-VAE effectively minimizes artifacts in complex dynamic scenarios. Video DC-AE extends the (Chen et al. 2024) framework for high-ratio video compression, achieving high-quality reconstruction. LTX-VAE achieves a high compression ratio of 1:192, preserving the ability to generate fine details. However, these models encounter issues during mobile deployment due to their high parameters and mismatched kernels. So we separately distill decoders for each model to improve inference speed while striving to maintain the original high quality.

4.2 Implementation Details

We train our Turbo-VAED on a subset of the VidGen (Tan et al. 2024) video dataset, consisting of 10k videos, which are preprocessed into 17-frame sequences at 256×256 resolution. We adopt the architecture from LTX-VAE as our initial decoder framework and refine it using the design techniques described in Section 3.2 and 3.3. Empirically, we set $\alpha_1 = 1.0$, $\alpha_2 = 1.0$, $\alpha_3 = 1 \times 10^{-7}$, and $\alpha_4 = 0.05$. The training is conducted on NVIDIA V100 GPUs, totaling about 300 GPU-hours, and gradient accumulation is implemented with an effective batch size of 32. We use AdamW optimizer with a learning rate of 2e-4 and β set to [0.9, 0.95].

4.3 Evaluation

Following (Seaweed et al. 2025; Wu et al. 2025a), we benchmark reconstruction quality on the UCF-101 (Soomro, Zamir, and Shah 2012) testval and DAVIS-2017 (Pont-Tuset et al. 2017) test datasets, reporting PSNR, LPIPS, SSIM, and reconstruction-FVD (rFVD) as evaluation metrics. We use the FVD metric to assess text-to-video generation performance on the OpenVid (Nan et al. 2024) dataset at 360×640 resolution. We report decoding latency on both the NVIDIA

Model	Decoder Param(M)	(d_t, d_h, d_w)	FPS@512×512↑		UCF-101@256×256				OpenVid FVD↓
			GPU	iPhone	PSNR↑	SSIM↑	LPIPS↓	rFVD↓	
HunyuanVideo	146.1	(4, 8, 8)	10.1	OOM	36.48	0.9663	0.0126	1.52	305.38
Turbo-VAED-Hunyuan	40.7	(4, 8, 8)	87.5	10.6	36.62	0.9674	0.0154	2.43	306.74
CogVideoX	123.4	(4, 8, 8)	10.6	OOM	36.23	0.9591	0.0197	4.73	254.67
Turbo-VAED-Cog	40.7	(4, 8, 8)	87.5	10.6	35.58	0.9606	0.0181	3.09	278.78
Video DC-AE	239.0	(4, 32, 32)	12.4	OOM	34.94	0.9594	0.0196	4.74	216.07
Turbo-VAED-DC	45.8	(4, 32, 32)	552.5	112.7	34.05	0.9475	0.0266	6.44	219.53
LTX Video	238.8	(8, 32, 32)	290.6	17.7	32.40	0.9192	0.0394	25.86	178.82
Turbo-VAED-LTX	41.9	(8, 32, 32)	841.6	161.8	31.68	0.9209	0.0419	25.01	178.69

Table 3: **Comparison with Recent Video VAEs.** We evaluate our proposed architecture (Sec. 3.2 3.3) and training strategy (Sec. 3.4) on four state-of-the-art video VAEs. Our method significantly reduces computational costs, with parameter counts reduced by up to 82.5%, effectively addressing the *OOM* issue, while preserving reconstruction and generation performance.

Model	Compression	FPS@512×512↑		FPS@720×1280↑		DAVIS@512×512		
		GPU	iPhone	GPU	iPhone	rFVD↓	PSNR↑	SSIM↑
SnapGen-V (Wu et al. 2025b)	1:192	—	31.5	—	—	—	—	—
H3AE (Wu et al. 2025a)	1:96	195.4	38.1	—	—	122.82	30.23	0.8412
Turbo-VAED-LTX	1:192	841.6	161.8	255.6	38.1	125.28	27.86	0.7905
Turbo-VAED-DC	1:96	552.5	112.7	167.0	25.3	49.91	30.08	0.8492

Table 4: **Comparison with Mobile-optimized VAEs.** Our models achieve significantly faster inference than prior mobile-optimized models while delivering competitive reconstruction quality.

Decoder Param(M)	Kernel Size	PSNR↑	LPIPS↓	SSIM↑
51.80	3	27.99	0.1310	0.8425
51.90	5	28.09	0.1285	0.8430
52.13	7	28.07	0.1307	0.8438

Table 5: **Ablation on 3D Convolution Kernel Size.** The 5×5 kernel size performs best.

A100 GPU and iPhone 16 Pro at 512px and 720p. All video datasets are used with 17 frames in standard settings and 16 frames for Turbo-VAED-DC during training and testing.

As shown in Table 3, the Turbo-VAED family retains quality with minimal degradation, and accelerates inference speed. Turbo-VAED-Hunyuan achieves $8.7\times$ speedups over Hunyuan-VAE for 512px video inference on GPUs, with slightly higher PSNR and SSIM than the original and minor trade-offs in LPIPS and FVD, demonstrating competitive reconstruction and generation performance. Similarly, Turbo-VAED-Cog delivers $8.2\times$ speedups at 512px compared to CogVideoX-VAE on GPUs while retaining comparable quality (8.1% lower LPIPS, 34.7% lower rFVD, 9.5% higher FVD). Both models enable mobile deployment at 512px resolution without OOM errors.

Turbo-VAED-DC delivers $44.4\times$ and $84.5\times$ speedups over Video DC-AE for 512px and 720p video inference

on GPUs, using just 19.2% of its parameters. At the same 1:96 compression ratio, Turbo-VAED-DC achieves a $2.9\times$ speedup in FPS over H3AE and demonstrates better reconstruction performance with a 59.4% reduction in rFVD (Table 4). While reducing parameters to 17.5%, Turbo-VAED-LTX delivers a $9.2\times$ speedup over LTX-VAE at 512px resolution on mobile devices, achieving comparable quality with slightly worse LPIPS but improved rFVD and FVD. For the first time, Turbo-VAED-DC and Turbo-VAED-LTX extend the capability of 720p video decoding to mobile devices, with Turbo-VAED-LTX achieving 38.1 FPS for this task.

4.4 Ablations

Ablation on 3D Convolution Kernel Size We perform ablation studies on 3D depthwise separable convolutions with different kernels. Larger kernels enhance model performance by expanding receptive fields, while depthwise separable convolutions reduce computational costs (Liu et al. 2022). As shown in Table 5, kernel sizes of $5 \times 5 \times 5$ and $7 \times 7 \times 7$ outperform the baseline, with the former achieving the best PSNR and LPIPS. Large kernels introduce a limited parameter increase and impose under 10 ms additional decoding latency on mobile devices. We adopt $5 \times 5 \times 5$ kernels for 3D depthwise separable convolutions in Turbo-VAED.

Ablation on Feature Alignment Location Aligning features on different decoder blocks impacts reconstruction quality. As shown in Table 6, aligning low-resolution features outperforms high-resolution counterparts, achieving a



prompt: A man in a dimly lit room talks on a vintage telephone, hangs up, and looks down with a sad expression. He holds the black rotary phone to his right ear with his right hand.

(a). Top: LTX Video. Bottom: Turbo-VAED-LTX.



prompt: A church is located on a rocky outcrop along a coast, showcasing its stunning architecture and the dramatic coastal views with crashing waves.

(b). Top: Video DC-AE. Bottom: Turbo-VAED-DC.



prompt: An elderly gentleman, with a serene expression, sits at the water's edge, a steaming cup of tea by his side. He holds brush in hand, as he renders an oil painting on a canvas.

(c). Top: CogVideoX. Bottom: Turbo-VAED-Cog.



prompt: A caravan of camels winds its way through endless golden dunes. The setting sun turns the desert a deep orange, and the sky is a gradient of purple and red.

(d). Top: Hunyuan Video. Bottom: Turbo-VAED-Hunyuan.

Figure 7: Text to Video Generation Results. The top and bottom rows display the generated videos, with latents produced by the original diffusion models and decoded via the original VAEs and their Turbo-VAED variants. These results with minimal visual differences demonstrate that Turbo-VAED preserves generation quality effectively.

Alignment Block	PSNR↑	LPIPS↓	SSIM↑
<i>mid</i>	26.30	0.1563	0.7972
<i>up</i> ₀	26.46	0.1514	0.8032
<i>up</i> ₁	26.42	0.1512	0.7992
<i>up</i> ₂	24.82	0.1837	0.7455
<i>up</i> ₀ & <i>up</i> ₁	<u>26.83</u>	<u>0.1441</u>	<u>0.8124</u>
<i>mid</i> & <i>up</i> ₀ & <i>up</i> ₁	26.91	0.1391	0.8155

Table 6: Ablation on Feature Alignment Location. Aligning multiple layers yields better reconstruction quality.

Projection Head	PSNR↑	LPIPS↓	SSIM↑
Linear	26.88	0.1470	0.8148
1-layer MLP	26.81	0.1424	0.8119
2-layer MLP	26.80	0.1445	0.8120
3D Pointwise Conv	26.91	0.1391	0.8155

Table 7: Ablation on Feature Projection Head. The two-layer 3D pointwise convolution network is the optimal choice.

17.7% improvement in LPIPS. Moreover, aligning multiple layers yields better results than any single-layer alignment, with an 8% LPIPS reduction compared to the best single-layer baseline. Empirically, these findings hold across all models in our experiments, leading us to adopt the multi-layer alignment strategy in all studies.

Ablation on Feature Projection Head We analyze the impact of different projection networks for feature alignment, as shown in Table 7. Feature alignment distillation employs a small projection head to project student features to match the teacher’s hidden dimension while providing extra flexibility (Bai et al. 2023). Observation indicates that a two-layer linear network built with $1 \times 1 \times 1$ convolutions outperforms other configurations.

This paper focuses on VAEs as deployment bottlenecks for video generative models on mobile devices. To address this problem, we propose a universal mobile-oriented video VAE decoder design, featuring (1) a parameter-efficient architecture based on 3D depthwise separable convolutions and (2) a decoupled 3D pixel shuffle upsampling strategy. We present a data-efficient training method enabling fast and stable transfer of video VAEs to mobile devices with negligible training cost. The solution is widely applicable to most video VAEs. It accelerates original VAEs by up to $84.5 \times$ at 720p resolution on GPUs, using as low as 17.5% of the original parameter count while preserving 96.9% of the original reconstruction quality. To our knowledge, Turbo-VAED achieves the first real-time 720p video VAE decoding on mobile devices. Our work aims to facilitate future research on the mobile deployment of large video generative models.

5 Conclusion

References

- Agarwal, N.; Ali, A.; Bala, M.; Balaji, Y.; Barker, E.; Cai, T.; Chattopadhyay, P.; Chen, Y.; Cui, Y.; Ding, Y.; et al. 2025. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*. 3
- Bai, Y.; Wang, Z.; Xiao, J.; Wei, C.; Wang, H.; Yuille, A. L.; Zhou, Y.; and Xie, C. 2023. Masked autoencoders enable efficient knowledge distillers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 24256–24265. 4, 7
- Bank, D.; Koenigstein, N.; and Giryes, R. 2023. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, 353–374. 3
- Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelovich, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*. 2, 3
- Chen, J.; Cai, H.; Chen, J.; Xie, E.; Yang, S.; Tang, H.; Li, M.; Lu, Y.; and Han, S. 2024. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*. 3, 5
- Cheng, Y.; and Yuan, F. 2025. LeanVAE: An Ultra-Efficient Reconstruction VAE for Video Diffusion Models. *arXiv preprint arXiv:2503.14325*. 3
- Doshi, D.; and Kim, J.-E. 2024. ReffAKD: Resource-efficient Autoencoder-based Knowledge Distillation. *arXiv preprint arXiv:2404.09886*. 4
- Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12873–12883. 3
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144. 3
- HaCohen, Y.; Chiprut, N.; Brazowski, B.; Shalem, D.; Moshe, D.; Richardson, E.; Levin, E.; Shiran, G.; Zabari, N.; Gordon, O.; et al. 2024. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*. 2, 4, 5
- Hansen-Estruch, P.; Yan, D.; Chung, C.-Y.; Zohar, O.; Wang, J.; Hou, T.; Xu, T.; Vishwanath, S.; Vajda, P.; and Chen, X. 2025. Learnings from Scaling Visual Tokenizers for Reconstruction and Generation. *arXiv preprint arXiv:2501.09755*. 3
- Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Movenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. 3
- Hu, S.; Tu, Y.; Han, X.; He, C.; Cui, G.; Long, X.; Zheng, Z.; Fang, Y.; Huang, Y.; Zhao, W.; et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*. 2, 3
- Kim, B.; Lee, K.; Jeong, I.; Cheon, J.; Lee, Y.; and Lee, S. 2025. On-device Sora: Enabling Training-Free Diffusion-based Text-to-Video Generation for Mobile Devices. *arXiv preprint arXiv:2503.23796*. 3
- Kingma, D. P.; Welling, M.; et al. 2013. Auto-encoding variational bayes. 2
- Kong, W.; Tian, Q.; Zhang, Z.; Min, R.; Dai, Z.; Zhou, J.; Xiong, J.; Li, X.; Wu, B.; Zhang, J.; et al. 2024. Hunyuan-video: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*. 2, 5
- Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986. 6
- Liu, Z.; Zhao, C.; Iandola, F.; Lai, C.; Tian, Y.; Fedorov, I.; Xiong, Y.; Chang, E.; Shi, Y.; Krishnamoorthi, R.; et al. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning*. 3
- Marafioti, A.; Zohar, O.; Farré, M.; Noyan, M.; Bakouch, E.; Cuenca, P.; Zakka, C.; Allal, L. B.; Lozhkov, A.; Tazi, N.; et al. 2025. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*. 2, 3
- Nan, K.; Xie, R.; Zhou, P.; Fan, T.; Yang, Z.; Chen, Z.; Li, X.; Yang, J.; and Tai, Y. 2024. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*. 5
- Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205. 2
- Peng, X.; Zheng, Z.; Shen, C.; Young, T.; Guo, X.; Wang, B.; Xu, H.; Liu, H.; Jiang, M.; Li, W.; et al. 2025. Open-sora 2.0: Training a commercial-level video generation model in \$200 k. *arXiv preprint arXiv:2503.09642*. 2, 5
- Polyak, A.; Zohar, A.; Brown, A.; Tjandra, A.; Sinha, A.; Lee, A.; Vyasa, A.; Shi, B.; Ma, C.; Chuang, C.; et al. 2024. Movie gen: A cast of media foundation models, 2025. URL <https://arxiv.org/abs/2410.13720>, 51. 3
- Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*. 5
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695. 2
- Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520. 3
- Seaweed, T.; Yang, C.; Lin, Z.; Zhao, Y.; Lin, S.; Ma, Z.; Guo, H.; Chen, H.; Qi, L.; Wang, S.; et al. 2025. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*. 5
- Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A. P.; Bishop, R.; Rueckert, D.; and Wang, Z. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874–1883. 4

- Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*. 5
- Tan, Z.; Yang, X.; Qin, L.; and Li, H. 2024. Vidgen-1m: A large-scale dataset for text-to-video generation. *arXiv preprint arXiv:2408.02629*. 5
- Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*. 2, 3
- Tian, R.; Dai, Q.; Bao, J.; Qiu, K.; Yang, Y.; Luo, C.; Wu, Z.; and Jiang, Y.-G. 2024. REDUCIO! Generating 1024×1024 Video within 16 Seconds using Extremely Compressed Motion Latents. *arXiv preprint arXiv:2411.13552*. 3
- Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 10347–10357. PMLR. 4
- Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30. 3
- Wu, Y.; Li, Y.; Skorokhodov, I.; Kag, A.; Menapace, W.; Girish, S.; Siarohin, A.; Wang, Y.; and Tulyakov, S. 2025a. H3AE: High Compression, High Speed, and High Quality AutoEncoder for Video Diffusion Models. *arXiv preprint arXiv:2504.10567*. 2, 3, 4, 5, 6
- Wu, Y.; Zhang, Z.; Li, Y.; Xu, Y.; Kag, A.; Sui, Y.; Coskun, H.; Ma, K.; Lebedev, A.; Hu, J.; et al. 2025b. SnapGen-V: Generating a five-second video within five seconds on a mobile device. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2479–2490. 2, 3, 6
- Xing, Y.; Fei, Y.; He, Y.; Chen, J.; Xie, J.; Chi, X.; and Chen, Q. 2024. Large Motion Video Autoencoding with Cross-modal Video VAE. *arXiv preprint arXiv:2412.17805*. 3
- Yahia, H. B.; Korzhenkov, D.; Lelekas, I.; Ghodrati, A.; and Habibian, A. 2024. Mobile Video Diffusion. *arXiv preprint arXiv:2412.07583*. 2, 3
- Yang, Z.; Li, Z.; Shao, M.; Shi, D.; Yuan, Z.; and Yuan, C. 2022a. Masked generative distillation. In *European conference on computer vision*, 53–69. Springer. 4
- Yang, Z.; Li, Z.; Zeng, A.; Li, Z.; Yuan, C.; and Li, Y. 2022b. Vitkd: Practical guidelines for vit feature knowledge distillation. *arXiv preprint arXiv:2209.02432*. 4
- Yang, Z.; Teng, J.; Zheng, W.; Ding, M.; Huang, S.; Xu, J.; Yang, Y.; Hong, W.; Zhang, X.; Feng, G.; et al. 2024. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*. 2, 5
- Yao, J.; Wang, C.; Liu, W.; and Wang, X. 2024. Fasteredit: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems*, 37: 56166–56189. 2
- Yao, J.; Yang, B.; and Wang, X. 2025. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 15703–15712. 3
- Ye, R.; Liu, F.; and Zhang, L. 2019. 3d depthwise convolution: Reducing model parameters in 3d vision tasks. In *Canadian Conference on Artificial Intelligence*, 186–199. Springer. 3
- Yu, L.; Lezama, J.; Gundavarapu, N. B.; Versari, L.; Sohn, K.; Minnen, D.; Cheng, Y.; Birodkar, V.; Gupta, A.; Gu, X.; et al. 2023. Language Model Beats Diffusion–Tokenizer is Key to Visual Generation. *arXiv preprint arXiv:2310.05737*. 3
- Zhao, S.; Zhang, Y.; Cun, X.; Yang, S.; Niu, M.; Li, X.; Hu, W.; and Shan, Y. 2024. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*. 3
- Zheng, Z.; Peng, X.; Yang, T.; Shen, C.; Li, S.; Liu, H.; Zhou, Y.; Li, T.; and You, Y. 2024. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*. 3