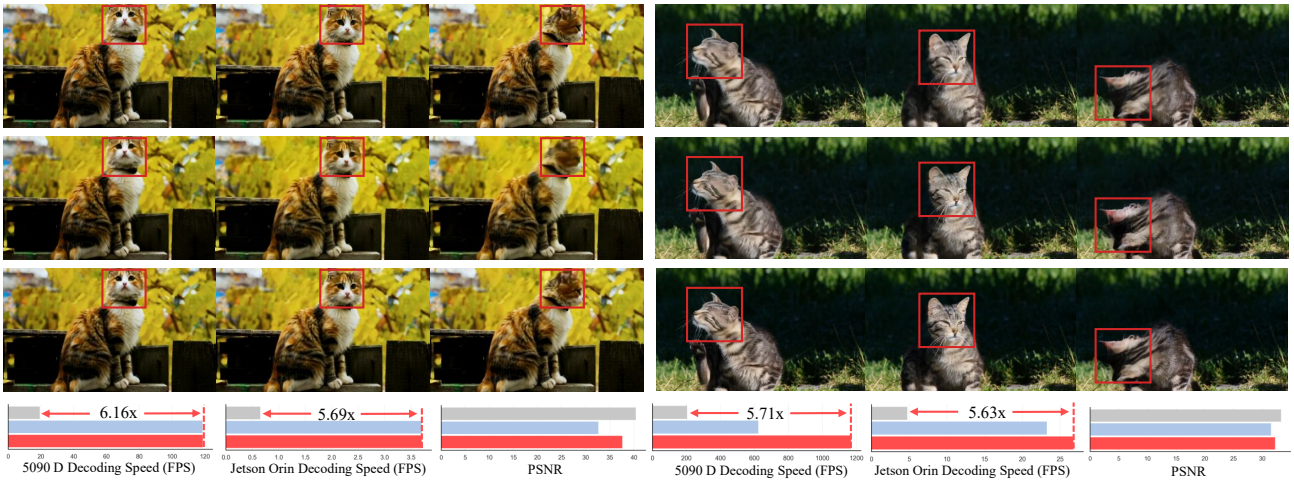
Flash-VAED: Plug-and-Play VAE Decoders for Efficient Video Generation

Anonymous Authors[1]



(a) Top: Wan 2.1 Middle: LightVAE-Wan 2.1 Bottom: Flash-VAED-Wan 2.1 (b) Top: LTX Video Middle: Turbo-VAED-LTX Bottom: Flash-VAED-LTX

Figure 1. **Qualitative and quantitative comparisons of video reconstruction results.** We evaluate Flash-VAED (Bottom) against the original VAE decoder (Top) and the current state-of-the-art baseline (Middle). Flash-VAED offers the fastest decoding speed with minimal loss of fidelity to the original VAE decoder.

Abstract

Latent diffusion models have enabled high-quality video synthesis, yet their inference remains costly and time-consuming. As diffusion transformers become increasingly efficient, the latency bottleneck inevitably shifts to VAE decoders. To reduce their latency while maintaining quality, we propose a universal acceleration framework for VAE decoders that preserves full alignment with the original latent distribution. Specifically, we propose (1) an *independence-aware channel pruning* method to effectively mitigate severe channel redundancy, and (2) a *stage-wise dominant operator optimization* strategy to address the high inference cost of the widely used causal 3D convolutions in VAE decoders. Based on these innovations, we construct a **Flash-VAED** family. Moreover, we design a *three-phase dynamic distillation* framework that efficiently transfers the capabilities of the original VAE decoder to Flash-VAED. Extensive experiments on Wan and LTX-Video VAE decoders demonstrate that our method outperforms baselines in both quality and speed, achieving approximately a **6× speedup** while maintaining the reconstruction performance up to **96.9%**. Notably, Flash-VAED accelerates the end-to-end generation pipeline by up to **36%** with negligible quality drops on VBench-2.0.

1. Introduction

Recently, artificial intelligence-generated content has witnessed remarkable breakthroughs across various domains, including text (Touvron et al., 2023; DeepSeek-AI et al., 2025), image (Labs, 2024; Xie et al., 2024), and video synthesis (HaCohen et al., 2024; Wan et al., 2025), achieving unprecedented levels of realism and coherence. The success of modern video generation models is largely attributed to powerful latent diffusion models (LDMs) (Ma et al., 2025), which typically comprise video variational auto-encoders (VAEs) (Kingma & Welling, 2013) and diffusion transformers (DiT) (Peebles & Xie, 2023).

Despite their exceptional performance, video generation

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

models are computationally expensive and slow in inference. For instance, it requires more than 30 minutes and 50 GB of GPU memory for the Wan 14B model to generate a 10-second 720p video clip on a standard H100 GPU. Such inefficiency significantly hinders practical deployment (Wu et al., 2025; Kim et al., 2025) and calls for more efficient video generation pipelines. Prior research has primarily focused on the DiT module, aiming to accelerate video generation by reducing denoising latency (Huang et al., 2025a; Zhang et al., 2025) or applying model compression (Guo et al., 2020; Wang et al., 2024; Huang et al., 2025b; Huang et al., 2025c). Nevertheless, as DiT acceleration techniques advance, the latency bottleneck has gradually shifted towards the VAE decoder, which has been largely overlooked.

In light of this issue, some recent studies have explored training lightweight VAEs from scratch (Cheng & Yuan, 2025; Li et al., 2025). However, these models often suffer from latent distribution misalignment with the original generation pipeline, incurring expensive fine-tuning on the DiT. In addition, some attempts have been made to structurally optimize the original VAE decoder (Zou et al., 2025; Contributors, 2025), yet they have neither fully investigated the underlying causes of the decoding latency bottleneck nor achieved an optimal trade-off between speed and quality.

In this work, we propose **Flash-VAED**, a family of efficient VAE decoders designed to accelerate video generation while preserving full alignment with the original latent distribution. Specifically, after a thorough analysis of the VAE decoder, we identify two primary factors that make it a latency bottleneck. First, through a singular value decomposition (SVD) analysis, we find that retaining merely $\sim$22% of the channels is sufficient to explain 99% of the variance in the full channel feature maps, which indicates severe channel redundancy. Based on this insight, we introduce an *independence-aware channel pruning* method that selectively optimizes a limited number of retained channels to linearly reconstruct the full channel feature maps. Second, we notice that causal 3D convolutions (CausalConv3D) dominate the latency within each decoder block. To address this issue, we analyze CausalConv3D across different stages and propose a *stage-wise dominant operator optimization* strategy, which replaces CausalConv3D with more efficient operators tailored to stage-specific characteristics. Moreover, to efficiently transfer the capabilities of the original VAE decoder to Flash-VAED, we develop a *three-phase dynamic distillation training framework*, achieving stage-wise alignment of Flash-VAED with the original VAE decoder.

In summary, our work aims to accelerate latent decoding for video generation and makes the following contributions:

- We propose an independence-aware channel pruning method, which reduces the channel count to 12.5% – 25% of the original with minimal quality loss.
- We introduce a stage-wise dominant operator optimization strategy that targets the dominant CausalConv3D, substituting it with efficient operators based on stage-specific characteristics to maximize efficiency.
- We present a three-phase dynamic distillation training framework that enables Flash-VAED to efficiently inherit the capabilities of the original VAE decoder.
- Extensive experiments on Wan and LTX-Video VAE decoders demonstrate that our method outperforms baselines in both quality and speed, achieving an approximate **6×** **speedup** while maintaining up to **96.9% performance** in reconstruction quality. Moreover, our method accelerates the entire generation pipeline by up to **36%** with negligible quality drops on VBench-2.0.

2. Related Work

Latent Diffusion Models. Diffusion models have achieved remarkable success in generative tasks, delivering unprecedented performance in video generation (HaCohen et al., 2024; Wan et al., 2025). Among recent advancements, architectures that employ DiT (Peebles & Xie, 2023) as the backbone have become particularly prominent, demonstrating superior scalability and generation quality compared to traditional U-Net-based designs (Ronneberger et al., 2015). However, performing the iterative diffusion process directly in the high-dimensional pixel space incurs prohibitive computational costs. To address this, VAEs have been introduced to compress raw data into compact low-dimensional latent representations. This paradigm, referred to as LDMs (Ma et al., 2025), significantly reduces inference complexity while maintaining high-fidelity synthesis. Nevertheless, as models scale up, both computational overhead and inference latency escalate substantially. To address this challenge, prior research has primarily targeted the DiT module, with strategies generally falling into two categories: reducing the number of denoising steps and reducing the latency per step. The former largely employs distillation techniques (Yin et al., 2024; Ge et al., 2025), while the latter mainly utilizes model compression methods such as pruning (Guo et al., 2020; Wang et al., 2024) and quantization (Shang et al., 2023; Huang et al., 2024; Huang et al., 2025c). Yet, as the efficiency of DiT improves, the latency bottleneck has gradually shifted towards the VAE decoder. For instance, when generating an 81-frame 480p video using the original pipeline for Wan 1.3B (Wan et al., 2025), VAE decoding accounts for only 2.3% of the total latency. However, with autoregressive few-step distillation (Huang et al., 2025a), this proportion surges to 29.6%, representing more than a ten-fold increase.

Video VAE Decoding Acceleration. Modern video VAE decoders (HaCohen et al., 2024; Kong et al., 2024; Wan et al., 2025) prioritize high-fidelity and consistent decoding by

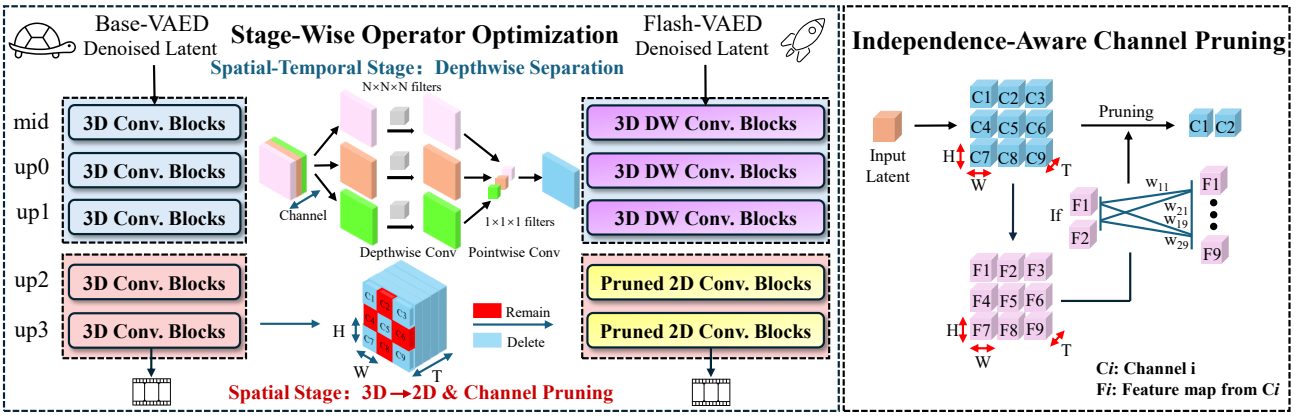


Figure 2. Overview of the Flash-VAED architecture[2]. The proposed stage-wise dominant operator optimization substitutes CausalConv3D with stage-specific efficient operators (left), tailored to each decoding stage. Moreover, the independence-aware channel pruning method (right) reduces the channel count to 12.5% − 25% of the original with minimal quality loss, leveraging channel independence.

employing computationally intensive CausalConv3D and a large number of channels. While beneficial for quality, these designs incur prohibitive computational and memory costs, hindering real-time deployment. To address these challenges, some studies focus on designing novel lightweight VAE architectures from scratch (Cheng & Yuan, 2025; Li et al., 2025). However, these approaches inevitably cause a latent distribution mismatch between the VAE decoder and the pre-trained DiT. Integrating them into the generation pipeline necessitates fine-tuning of the DiT, which incurs prohibitive computational costs. Alternatively, another line of research attempts to perform structural optimizations directly on the original VAE decoder to maintain latent consistency (Zou et al., 2025; Contributors, 2025). However, these methods neither fully address the root causes of the latency bottleneck in VAE decoders nor achieve an optimal trade-off between speed and quality. For instance, Turbo-VAED (Zou et al., 2025) addresses the parameter redundancy of CausalConv3D in low-resolution layers but fails to resolve the significant inference latency in high-resolution layers. Similarly, LightVAE (Contributors, 2025) identifies channel redundancy yet fails to effectively reduce it without incurring quality loss. In contrast, our work comprehensively identifies the underlying causes of the decoding latency bottleneck, achieving substantial acceleration with negligible quality degradation.

3. Method

In this section, we analyze the two primary factors that contribute to the latency bottleneck in the VAE decoder for video generation and present our corresponding solutions. The first factor is *severe channel redundancy*, which we address by proposing independence-aware channel pruning, as illustrated on the right side of Figure 2 and detailed in Section 3.1. The second factor is the *ubiquitous presence and high inference cost of CausalConv3D*. To address this issue, we introduce a stage-wise dominant operator optimization strategy, as shown on the left side of Figure 2, with further details provided in Section 3.2. Building on these optimizations, we construct the **Flash-VAED** family. Finally, to efficiently transfer the capabilities of the original VAE decoder to Flash-VAED and seamlessly embed Flash-VAED into the original generation pipeline, we design a comprehensive three-stage feature distillation training framework, which is elaborated in Section 3.3.

For clarity, we standardize the naming of the decoder blocks. From input to output, corresponding to low to high resolutions, we refer to the main blocks as the Middle Block, Upsample Block 0, Upsample Block 1, Upsample Block 2, and Upsample Block 3, denoted by mid, up_0, up_1, up_2, and up_3, respectively.

3.1. Independence-Aware Channel Pruning

Channel pruning is a widely used neural network acceleration technique (Liu et al., 2017; Cheng et al., 2024). However, applying conventional pruning methods to video VAE decoders faces several challenges. First, our visualization of the output feature maps and the calculation of the cosine similarity against Channel 0, as shown in Figure 3, reveal observable redundancy. Nevertheless, pairwise similarity measures fail to provide sufficient evidence for pruning decisions. Second, the extensive use of cascaded residual blocks in the decoder complicates the pruning process, as maintaining identical channel indices across different blocks is critical to preserve the model's original connectivity. A detailed explanation is provided in Appendix A.2.

To address these challenges, we shift from pairwise similarity to linear dependence, where a channel is considered redundant if it can be linearly represented by other channels. To validate this concept, we conduct an SVD analysis on the matrix of feature maps across all channels. As illustrated

[2]We take Flash-VAED-Wan 2.1 as an example.

in Figure 4, we find that retaining only **22%** of the singular values is sufficient to explain **99%** of the total variance. Based on this insight, we propose an independence-aware channel pruning method that reduces the channel count to 12.5%−25% of the original size with minimal quality loss, while maintaining the internal continuity of the model.

Mathematically, we use $\mathbf{X}$ to denote the retained channel features and linearly reconstruct the full channel features $\mathbf{Y}$ as follows (Luo et al., 2017):

$$\hat{\mathbf{Y}} = \mathbf{W}\mathbf{X}, \tag{1}$$

where the projection matrix $\mathbf{W}$ is derived using the least squares method (Björck, 1990). Since traditional metrics for quantifying the reconstruction fidelity, such as mean squared error, exhibit inherent sensitivity to feature magnitude variations, we instead employ the coefficient of determination R^2 to measure how well the retained channel features explain the original full channel features (Lai et al., 2018):

$$\mathrm{R}^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}} = 1 - \frac{\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|_F^2}, \tag{2}$$

where $\mathrm{SS}_{\mathrm{res}}$ and $\mathrm{SS}_{\mathrm{tot}}$ denote the residual and total sum of squares, respectively, and $\|\cdot\|_F$ is the Frobenius norm.

Based on this foundation, our independence-aware channel pruning framework unfolds in three integrated steps. First, we employ a greedy channel selection strategy to iteratively identify the optimal subset of channels. Specifically, given the current subset, the next channel is selected by ensuring that each addition maximizes the marginal gain of R^2 until the target number of channels is reached. Second, to ensure that the selected channels adequately represent the full feature map, we introduce a pre-pruning retained channel enhancement. We encourage the model to maximize the expressive power of the retained channels by incorporating an expressivity loss during training with respect to $\mathbf{W}$ and $\mathbf{X}$, which is formulated as:

$$\mathrm{L}_{\mathrm{ce}} = 1 - \mathrm{R}^2 = \frac{\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|_F^2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|_F^2}. \tag{3}$$

As shown in Figure 5, this operation significantly enhances the reconstruction capability, with R^2 scores improving from an initial range of 0.87−0.93 to consistently exceeding 0.98 across all layers. Finally, to resolve the disruption of residual block continuity caused by mismatched retained indices across blocks, we implement a topology-preserving shortcut injection. Specifically, standard identity shortcuts are replaced with 1×1 convolutions. Since retained channels are capable of linearly representing any channel within the current block, we can directly derive the mapping matrix $\mathbf{W}$ from the retained indices of the current block to the retained indices of the subsequent block using the least squares method. By initializing the 1×1 convolution shortcut of the subsequent block with this matrix $\mathbf{W}$, we effectively preserve the internal continuity of the pruned VAE decoder. More details are provided in Appendix A.2.

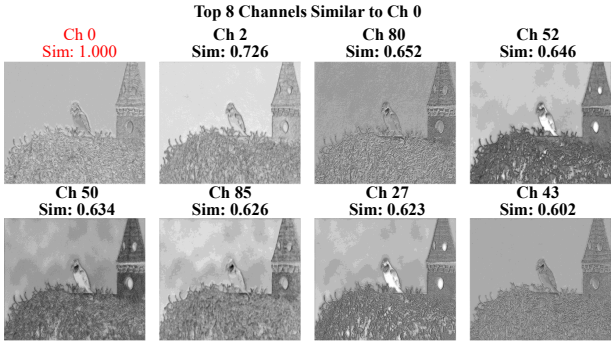


Figure 3. **Channel-wise similarity analysis.** We visualize the top-8 channels most similar to Channel 0. Although the feature maps exhibit visual similarity, the quantitative similarity scores are **not high enough** to support pruning.

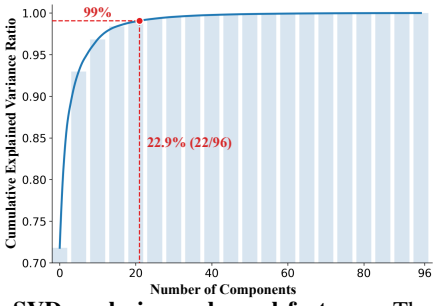


Figure 4. **SVD analysis on channel features.** The curve for cumulative explained variance ratio reveals the intrinsic low-rank nature of the feature maps. Notably, only 22 components (22.9% of the total) are required to explain 99% of the feature variance, providing strong empirical support for our pruning method.

3.2. Stage-Wise Dominant Operator Optimization

In addition to channel redundancy, our comprehensive latency analysis identifies CausalConv3D as another key computational bottleneck within the VAE decoder. As shown in Figure 6, this operator consistently consumes over **60%** of the inference time across most blocks, with computational costs escalating dramatically in high-resolution stages. Therefore, we characterize CausalConv3D as the dominant operator. To tackle this, we propose a stage-wise replacement strategy that substitutes CausalConv3D with tailored efficient operators based on the specific characteristics of each decoding stage. This approach achieves significant reductions in both parameter count and inference latency with negligible quality degradation.

Specifically, guided by insights from previous works (Zou et al., 2025; Sandler et al., 2018; Howard et al., 2017), we start from the deep layers (low-resolution layers). In these layers, we substitute CausalConv3D with 3D depthwise separable convolutions (3D DW Conv.), reducing the parameter count to approximately 20% of the original VAE decoder with minimal quality degradation. The theoretical analysis is provided in Appendix A.1. For the shallow layers (high-resolution layers), since temporal upsampling has been completed in deep layers, we hypothesize that the reliance on inter-frame temporal dependencies diminishes

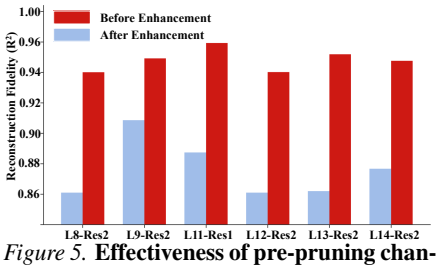


Figure 5. **Effectiveness of pre-pruning channel enhancement.** We compare the reconstruction fidelity R^2 of the retained channels before and after enhancement. The significant improvement across all layers validates that our strategy effectively forces the retained channels to encode more information.

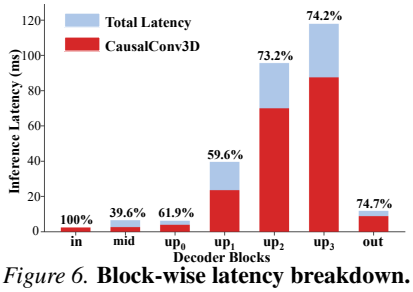


Figure 6. **Block-wise latency breakdown.** The CausalConv3D operator occupies over **60%** of the total latency in most blocks, and its inference time grows rapidly with resolution. These observations jointly lead us to define it as the **dominant operator**.

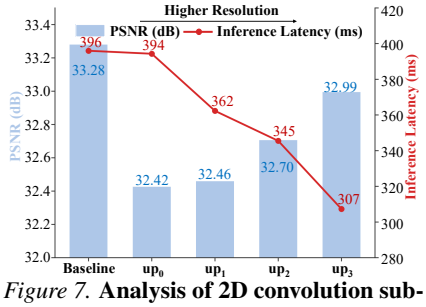


Figure 7. **Analysis of 2D convolution substitution.** Moving towards higher resolution layers, replacing CausalConv3D with 2D convolutions consistently reduces the latency while delivering less quality loss.

here. Consequently, computationally expensive 3D operations can be safely replaced by spatial-only operators. To validate this hypothesis, we conduct an experiment by progressively replacing CausalConv3D with 2D convolutions from deep to shallow layers. As illustrated in Figure 7, a clear trend emerges: as the replacement shifts toward higher-resolution layers, the latency increasingly reduces, while the quality loss diminishes. These empirical results strongly support our strategy to replace CausalConv3D in the shallow layers with lightweight 2D convolutions for faster inference.

3.3. Training Strategy: Three-Phase Dynamic Distillation Framework

To enable **Flash-VAED** to efficiently inherit the capabilities of the original VAE decoder, we incorporate feature distillation into our training objective. Following the paradigm established in prior works (Bai et al., 2022; Doshi & Kim, 2024), we define a feature-based distillation loss as follows:

$$\mathrm{L}_{\mathrm{distill}} = \sum_{l} \frac{1}{\mathrm{numel}(f_l^F)} \sum_{i} \left\| \sigma(f_l^F)_i - f_{l,i}^O \right\|_1, \quad \text{with} \quad \sigma(\cdot) = \begin{cases} \mathrm{Identity}(\cdot), & \text{phase 1 \& 2,} \\ \mathrm{Conv}_{1\times1}(\cdot), & \text{phase 3,} \end{cases} \tag{4}$$

where f_l^F and f_l^O denote the intermediate feature maps extracted from **Flash-VAED** and the original VAE decoder in block l, respectively, $\mathrm{numel}(\cdot)$ is the normalization factor that represents the total element count, and $\sigma(\cdot)$ serves as a phase-dependent adaptation layer that aligns the features of Flash-VAED with the original feature space.

Based on this, we propose a three-phase dynamic distillation training strategy. In phase 1, we aim to capture global structural information by directly aligning the features of the deep layers of Flash-VAED with those of the original VAE decoder. The complete loss function is formulated as

$$L = \alpha_1 L_1 + \alpha_2 L_{\mathrm{lpips}} + \alpha_3 L_{\mathrm{distill}} + \alpha_4 L_{\mathrm{ssim}}. \tag{5}$$

Then, phase 2 focuses on enhancing pre-pruned channels by incorporating L_{ce} from Equation 3 into the overall objective to maximize the expressivity of retained channels. The complete loss function for this phase is represented as

$$L = \alpha_1 L_1 + \alpha_2 L_{\mathrm{lpips}} + \alpha_3 L_{\mathrm{distill}} + \alpha_4 L_{\mathrm{ssim}} + \alpha_5 L_{\mathrm{ce}}. \tag{6}$$

Finally, phase 3 targets fine-grained recovery in pruned shallow layers. To address channel number misalignment caused by pruning, we instantiate σ as a 1×1 convolution. Notably, we initialize this projection layer using $\mathbf{W}$ derived in Section 3.1 to significantly accelerate convergence. The loss function for this phase is the same as Equation 5.

4. Experiments

In this section, we first describe the experiment setup, including the implementation details of our Flash-VAED family, training and benchmark datasets, baselines, and evaluation protocols in Section 4.1. Then, we present our main results regarding both quality and efficiency in Section 4.2. Finally, we conduct ablation studies to justify the design choices for critical components and parameters in Section 4.3.

4.1. Experimental Setup

Flash-VAED Family. To demonstrate the versatility and robustness of our approach, we instantiate the Flash-VAED family by applying our optimization methods to the decoders of two representative state-of-the-art (SOTA) video VAEs: Wan (Wan et al., 2025) and LTX-Video (HaCohen et al., 2024). The rationale for this selection is twofold. First, we choose Wan (Wan et al., 2025) because its VAE decoder is widely recognized for delivering the highest quality among mainstream video generation models, serving as a benchmark for high-fidelity performance. Second, LTX-Video (HaCohen et al., 2024), which operates at an extreme spatial-temporal compression ratio of 1:192, demonstrates the superior efficiency and effectiveness of our method even under ultra-high compression regimes.

Datasets. To ensure robustness for both reconstruction and

Table 1. **Quantitative comparison of video reconstruction results with SOTA methods.** We compare Flash-VAED against the original VAE decoders and competitive baselines on both a consumer-grade GPU (RTX 5090D) and an edge device (Jetson Orin), where d_T, d_H, and d_W denote the compression ratios for time, height, and width, respectively, ↑ indicates higher is better, while ↓ indicates lower is better. The best results among the methods are highlighted in **bold**. Flash-VAED outperforms all the baselines in both speed and quality.

Model	(d_T, d_H, d_W)	FPS ↑		Reconstruction Quality		
		RTX 5090D	Jetson Orin	PSNR ↑	SSIM ↑	LPIPS ↓
Wan 2.1 (Wan et al., 2025)	$(4, 8, 8)$	19.27	0.65	40.40	0.9733	0.0190
LightVAE-Wan 2.1 (Contributors, 2025)	$(4, 8, 8)$	118.60	**3.70**	32.61	0.9416	0.0892
Flash-VAED-Wan 2.1 (Ours)	$(4, 8, 8)$	**118.77**	**3.70**	**37.61**	**0.9614**	**0.0285**
LTX-Video (HaCohen et al., 2024)	$(8, 32, 32)$	204.55	4.75	33.28	0.9253	0.0497
Turbo-VAED-LTX (Zou et al., 2025)	$(8, 32, 32)$	623.08	23.24	31.52	0.9275	0.0555
Flash-VAED-LTX (Ours)	$(8, 32, 32)$	**1167.99**	**26.74**	**32.24**	**0.9293**	**0.0551**

generation tasks, we curate a hybrid training dataset that comprises 10K samples, equally drawn from two distinct sources. The first half consists of 5K video-latent pairs obtained through video reconstruction, where videos sampled from the VidGen (Tan et al., 2024) dataset and latents are extracted using the original VAE encoder. The second half comprises 5K latent-video pairs generated via a pipeline where diverse text prompts generated by GPT-4o are fed into the video generation model (Wan et al., 2025; HaCohen et al., 2024) to collect pairs of final denoised latents and generated videos. For quantitative evaluation, following prior work (Zou et al., 2025), we utilize the UCF-101 test set (Soomro et al., 2012) as our reconstruction benchmark.

Baselines. We compare our Flash-VAED against the following baselines in terms of both quality and efficiency:

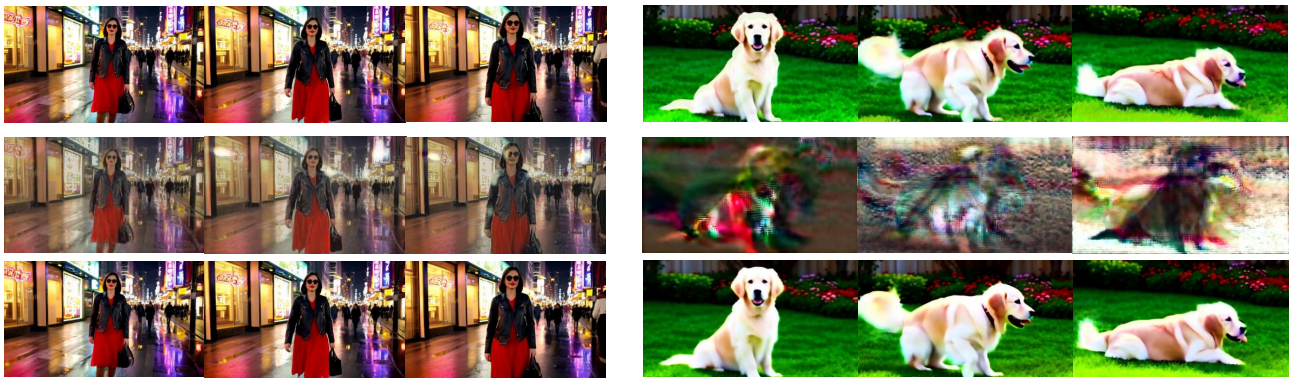
- **Turbo-VAED** (Zou et al., 2025): The current SOTA framework for accelerating VAE decoders specifically for video generation tasks.
- **LightVAE** (Contributors, 2025): An open-source solution in the Hugging Face LightX2V series, known for its strong practical adoption, with over 7,000 monthly downloads.

Metrics. To comprehensively evaluate our method, we first assess the reconstruction quality using three standard metrics: PSNR for pixel-level fidelity, SSIM for structural preservation, and LPIPS for perceptual realism aligned with human vision. To rigorously verify Flash-VAED's capability on compute-constrained devices, we benchmark inference speed (FPS) on both a consumer-grade GPU (NVIDIA RTX 5090D) and an edge device (NVIDIA Jetson AGX Orin 64G) (Karumbunathan, 2022). For the generation task, we measure end-to-end latency to evaluate efficiency. The quality evaluation is conducted using VBench 2.0 (Zheng et al., 2025), the most authoritative and comprehensive benchmark for assessing models across 18 distinct dimensions, ensuring that our acceleration maintains high-quality generation standards with holistic evaluation.

Implementation Details. Flash-VAED is trained on video sequences with a resolution of 480×832 spanning 81 frames. The training process is conducted on four NVIDIA A5880 GPUs and requires approximately 300 GPU-hours. We employ the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4}. The loss coefficients are set as $\alpha_1 = 10$, $\alpha_2 = 2$, $\alpha_3 = 1$, and $\alpha_4 = 5$ during the first and third training stages, while in the second stage, we maintain these values and additionally set $\alpha_5 = 1$. In terms of specific architectural adaptations, for the Wan 2.1 VAE decoder (Wan et al., 2025), we replace the blocks mid, up_0, and up_1 with depthwise separable convolutions, and switch up_2 and up_3 to 2D convolutions, applying $1/8$ channel pruning to the latter two blocks. Similarly, for the LTX-Video VAE decoder (HaCohen et al., 2024), we replace the blocks mid, up_0, and up_1 with depthwise separable convolutions, keep up_2 unchanged, replace up_3 with a 2D convolution, and apply $1/4$ channel pruning to both up_2 and up_3.

4.2. Main Results

Reconstruction Results. We present qualitative visual results in Figure 1 and report quantitative evaluations in Table 1. For the Wan 2.1 VAE decoder (Wan et al., 2025), Flash-VAED achieves substantial speedups of 6.16× on the NVIDIA RTX 5090 D and 5.69× on the Jetson Orin, while retaining 93.1% of the original VAE decoder's quality. Notably, our reconstructed PSNR reaches 37.61 dB, surpassing even the original VAE decoders of other SOTA video generation models (Kong et al., 2024; HaCohen et al., 2024; Yang et al., 2024), which demonstrates the effectiveness of our method on VAE decoders with mainstream compression ratios. Compared to the baseline LightVAE-Wan2.1 (Contributors, 2025), Flash-VAED achieves an impressive 5 dB gain in PSNR at the same inference speed. For the LTX-Video VAE (Contributors, 2025), Flash-VAED achieves 5.71× and 5.63× speedups on the NVIDIA RTX 5090 D and Jetson Orin, respectively, while preserving 96.9% of the original quality. This further validates the efficacy of our method on VAE decoders with high compression ratios. Compared with Turbo-VAED (Zou et al., 2025), Flash-VAED outperforms in both speed and quality. Specifically, Flash-VAED is nearly 2× faster than Turbo-VAED on the NVIDIA RTX



Prompt example: A stylish woman strolls down a bustling Tokyo street, the warm glow of neon lights and animated city signs casting vibrant reflections. She wears a sleek black leather jacket paired with a flowing red dress and black boots, her black purse slung over her shoulder.

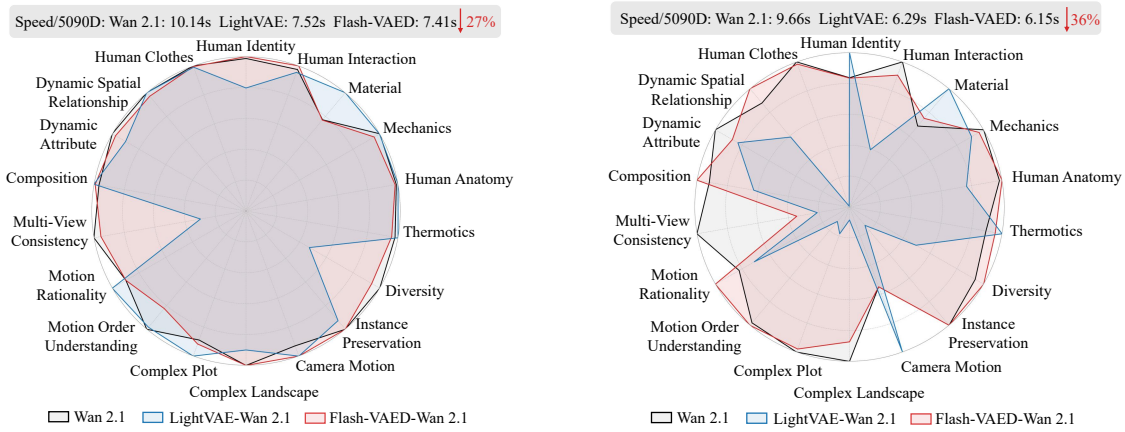
(a) Self Foring-Wan 1.3B

Prompt example: A fluffy golden retriever sits calmly in a sunlit backyard, surrounded by vibrant green grass and colorful flower beds, its eyes alert and ears perked. Suddenly, with a burst of energy, it leaps up and begins to run in joyful circles, tail wagging furiously.

(b) FastVideo-Wan 1.3B

Top: Wan 2.1 VAE Middle: LightVAE-Wan 2.1 Bottom: Flash-VAED-Wan 2.1

Figure 8. **Visual comparison of video generation results.** Our Flash-VAED (bottom) matches closely with the original Wan 2.1 VAE (top) in fidelity, details and textures, whereas the LightVAE (middle) exhibits severe artifacts that results in invalid video content. The full prompts and additional details are provided in Appendix A.4.



(a) Self Forcing-Wan 1.3B *(b)* FastVideo Wan-1.3B

Figure 9. Generation performance of Flash-VAED embedded on the VBench-2.0. We employ Flash-VAED to replace the VAE decoders of: (a) the Self Forcing-Wan 1.3B, and (b) the FastVideo-Wan 1.3B, whose overall generation quality is evaluated on VBench-2.0.

5090 D and even faster on the Jetson Orin, without specific edge optimizations. For quality comparison, our proposed Flash-VAED surpasses Turbo-VAED across all PSNR, SSIM, and LPIPS metrics. These superiority arises from (i) a lightweight model design that facilitates fast inference, and (ii) an efficient training strategy that enables Flash-VAED to retain the capabilities of the original VAE decoders.

Generation Results. To demonstrate the versatility of our method, we integrate Flash-VAED into two representative accelerated generation frameworks: Self Forcing-Wan 1.3B, an autoregressive acceleration framework (Huang et al., 2025a), and FastVideo-Wan 1.3B, a few-step distillation acceleration framework (Zhang et al., 2025). The video generation results are qualitatively visualized in Figure 8 for Self Forcing-Wan 1.3B (left) and FastVideo-Wan 1.3B (right), with the corresponding quantitative results evaluated

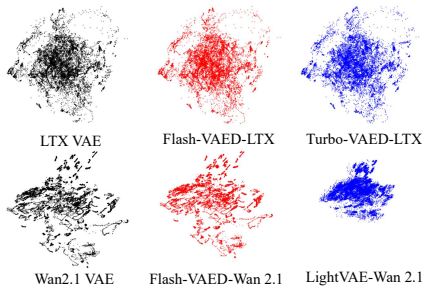


Figure 10. Visualization of samples from the latent distribution.

on VBench 2.0 reported in Figure 9 on the left and right sides, respectively. For the Self Forcing pipeline (Huang et al., 2025a), Flash-VAED achieves a 27% speedup in end-to-end generation latency, while the performance curve in Figure 9a closely tracks that of the original VAE decoder across all dimensions in VBench-2.0 (Zheng et al., 2025).

In contrast, the baseline LightVAE (Contributors, 2025), although comparable (slightly slower) in speed, demonstrates noticeable performance drops in several dimensions. The advantages of Flash-VAED are even more pronounced in the FastVideo pipeline (Zhang et al., 2025), where it achieves a 36% speedup while maintaining a closely aligned performance curve with the original VAE decoder. Conversely, the performance of the LightVAE-Wan 2.1 baseline degrades significantly, resultining in the decoding of latent information into meaningless noise. This disparity comes from the mismatch in latent distribution, as illustrated in Figure 10, which visualizes the latent distribution of the original VAE, Flash-VAED, and the baseline. It is evident that the latent distribution of Flash-VAED (red) perfectly overlaps with that of the original VAE decoder (black), ensuring seamless compatibility with the generation pipeline, while LightVAE (blue) (Contributors, 2025) suffers from a significant distributional deviation.

4.3. Ablation Study

Ablation on Pruning Methods. To validate our independence-aware channel pruning strategy, we compare it against a random pruning approach. As shown in Figure 11, our method demonstrates a significantly higher starting point, faster performance growth, and superior outcomes. These improvements verify that our independence-aware channel selection effectively preserves the most informative channels, laying a solid foundation for the retained-channel enhancement in the second stage, while ensuring that the retained channels effectively inherit the capabilities from the original full channel set.

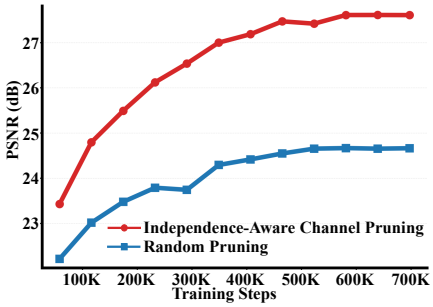


Figure 11. Ablation results of using different pruning methods.

Ablation on Pruning Ratios. Taking the LTX-Video VAE decoder (HaCohen et al., 2024) as a representative example, we investigate the speed-quality trade-off across different pruning ratios $(1, 1/2, 1/4, 1/8)$, as presented in Table 2. As the pruning ratio decreases from 1 to $1/8$, the inference speed steadily increases, with a degradation in performance. We select a 1/4 ratio for the LTX-Video VAE decoder (HaCohen et al., 2024) as it achieves the best trade-off: compared with a $1/2$ ratio, the $1/4$ ration provides a $2\times$ speedup with only a 0.93% drop in quality, while the $1/8$ ratio incurs a more substantial 3.7% quality loss. Similarly, for the Wan 2.1 VAE decoder (Wan et al., 2025), we identify the $1/8$ ratio as the optimal configuration that effectively balances speed and quality.

Table 2. Ablation results of using different pruning ratios.

Prune Ratio	FPS/5090D ↑	PSNR ↑	SSIM ↑	LPIPS ↓
1	204.55	33.2883	0.9253	0.0497
1/2	625.00	32.5421	0.9307	0.0521
1/4	1167.99	32.2400	0.9293	0.0551
1/8	1907.23	31.0786	0.9162	0.0804

Ablation on Trainning Strategies. To validate the rationale behind our three-phase dynamic distillation pipeline, we compare our method against settings without distillation and those applying distillation exclusively to shallow or deep layers. As shown in Table 4, our method maximizes the transfer of capabilities from the original VAE decoders to Flash-VAED, achieving 96.9% of the original quality. Moreover, we evaluate the impact of initializing the projection layer using the W matrix derived from the second training stage during the third training stage, comparing it against random initialization. As indicated in Table 3, our initialization method leads to approximately a 6.25% improvement in reconstruction quality.

Table 3. Ablation results of different projected layer initialization.

Project Layer	PSNR ↑	SSIM ↑	LPIPS ↓
Random Initialization	30.3061	0.9013	0.0978
Ours	**32.2400**	**0.9293**	**0.0551**

Table 4. Ablation results of using different distillation methods.

Training Methods	PSNR ↑	SSIM ↑	LPIPS ↓
Without Distillation	30.7957	0.9125	0.0831
Deep-only Distillation	30.8065	0.9126	0.0830
Shallow-only Distillation	31.2386	0.9152	0.0722
Ours	**32.3768**	**0.9317**	**0.0566**

5. Conclusions

In this paper, we focus on accelerating VAE decoders for video generation while ensuring strict alignment with the original latent distribution. We first experimentally identify two primary factors that contribute to the latency bottleneck of VAE decoders: extreme channel redundancy and the ubiquitous presence and high inference cost of CausalConv3D operations. To address these challenges, we propose two targeted strategies: independence-aware channel pruning and stage-wise dominant operator optimization. Building on these optimizations, we introduce the Flash-VAED family. Furthermore, to facilitate the efficient transfer of capabilities from the original VAE decoders to Flash-VAED, we develop a three-phase dynamic distillation training pipeline. Extensive experiments demonstrate that Flash-VAED significantly outperforms baselines, achieving the fastest decoding speed with minimal loss of fidelity to the original VAE decoders.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Bai, Y., Wang, Z., Xiao, J., Wei, C., Wang, H., Yuille, A., Zhou, Y., and Xie, C. Masked autoencoders enable efficient knowledge distillers, 2022. URL https://arxiv.org/abs/2208.12256.

Björck, Å. Least squares methods. *Handbook of numerical analysis*, 1:465–652, 1990.

Cheng, H., Zhang, M., and Shi, J. Q. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Cheng, Y. and Yuan, F. Leanvae: An ultra-efficient reconstruction vae for video diffusion models, 2025. URL https://arxiv.org/abs/2503.14325.

Contributors, L. Lightx2v: Light video generation inference framework. https://github.com/ModelTC/lightx2v, 2025.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu, J., Li, J., Song, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang, T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Song, X., Shan, X., Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhu, Y. X., Zhang, Y., Xu, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y., Yan, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Huang, Z., Zhang, Z., Xie, Z., Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu, Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Gao, Z., and Pan, Z. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Doshi, D. and Kim, J.-E. Reffakd: Resource-efficient autoencoder-based knowledge distillation. *arXiv preprint arXiv:2404.09886*, 2024.

Ge, X., Zhang, X., Xu, T., Zhang, Y., Zhang, X., Wang, Y., and Zhang, J. Senseflow: Scaling distribution matching for flow-based text-to-image distillation, 2025. URL https://arxiv.org/abs/2506.00523.

Guo, J., Ouyang, W., and Xu, D. Multi-dimensional pruning: A unified framework for model compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1508–1517, 2020.

HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., Panet, P., Weissbuch, S., Kulikov, V., Bitterman, Y., Melumian, Z., and Bibi, O. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Huang, X., Li, Z., He, G., Zhou, M., and Shechtman, E. Self forcing: Bridging the train-test gap in autoregressive video diffusion, 2025a. URL https://arxiv.org/abs/2506.08009.

Huang, Y., Gong, R., Liu, J., Chen, T., and Liu, X. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7362–7371, 2024.

Huang, Y., Gong, R., Liu, J., Ding, Y., Lv, C., Qin, H., and Zhang, J. Qvgen: Pushing the limit of quantized video generative models, 2025b. URL https://arxiv.org/abs/2505.11497.

Huang, Y., Gong, R., Liu, X., Liu, J., Li, Y., Lu, J., and Tao, D. Temporal feature matters: A framework for diffusion model quantization, 2025c. URL https://arxiv.org/abs/2407.19547.

Karumbunathan, L. S. Nvidia jetson agx orin series. *A Giant Leap Forward for Robotics and Edge AI Applications. Technical Brief*, 2022.

Kim, B., Lee, K., Jeong, I., Cheon, J., Lee, Y., and Lee, S. On-device sora: Enabling training-free diffusion-based text-to-video generation for mobile devices. *arXiv preprint arXiv:2502.04363*, 2025.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Labs, B. F. Flux. https://github.com/black-forest-labs/flux, 2024.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Li, Z., Lin, B., Ye, Y., Chen, L., Cheng, X., Yuan, S., and Yuan, L. Wf-vae: Enhancing video vae by wavelet-driven energy flow for latent video diffusion model, 2025. URL https://arxiv.org/abs/2411.17459.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Luo, J.-H., Wu, J., and Lin, W. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Ma, X., Wang, Y., Chen, X., Jia, G., Liu, Z., Li, Y.-F., Chen, C., and Qiao, Y. Latte: Latent diffusion transformer for video generation, 2025. URL https://arxiv.org/abs/2401.03048.

Peebles, W. and Xie, S. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Shang, Y., Yuan, Z., Xie, B., Wu, B., and Yan, Y. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1972–1981, 2023.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. URL https://arxiv.org/abs/1212.0402.

Tan, Z., Yang, X., Qin, L., and Li, H. Vidgen-1m: A large-scale dataset for text-to-video generation, 2024. URL https://arxiv.org/abs/2408.02629.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.-F., and Liu, Z. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Wang, K., Chen, J., Li, H., Mi, Z., and Zhu, J. Sparsedm: Toward sparse efficient diffusion models. *arXiv preprint arXiv:2404.10445*, 2024.

Wu, Y., Zhang, Z., Li, Y., Xu, Y., Kag, A., Sui, Y., Coskun, H., Ma, K., Lebedev, A., Hu, J., et al. Snapgen-v: Generating a five-second video within five seconds on a mobile device. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2479–2490, 2025.

Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., and Han, S. Sana: Efficient high-resolution image synthesis with linear diffusion transformers, 2024. URL https://arxiv.org/abs/2410.10629.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., and Freeman, W. T. Improved distribution matching distillation for fast image synthesis, 2024. URL https://arxiv.org/abs/2405.14867.

Zhang, P., Huang, H., Chen, Y., Lin, W., Liu, Z., Stoica, I., Xing, E., and Zhang, H. Vsa: Faster video diffusion with trainable sparse attention. *arXiv preprint arXiv:2505.13389*, 2025.

Zheng, D., Huang, Z., Liu, H., Zou, K., He, Y., Zhang, F., Gu, L., Zhang, Y., He, J., Zheng, W.-S., Qiao, Y., and Liu, Z. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness, 2025. URL https://arxiv.org/abs/2503.21755.

Zou, Y., Yao, J., Yu, S., Zhang, S., Liu, W., and Wang, X. Turbo-vaed: Fast and stable transfer of video-vaes to mobile devices, 2025. URL https://arxiv.org/abs/2508.09136.

A. Implementation Details

A.1. Mechanism of Depthwise Separable Convolutions

Standard 3D convolutions effectively model spatiotemporal correlations (across time, height, and width) and cross-channel correlations. While effective, this approach incurs high computational costs. To address this, depthwise separable 3D convolution decomposes this operation into two efficient stages: spatiotemporal filtering and channel mixing. We use T, H, and W to denote the number of frames, the height, and the width of the videos, respectively, while C_{in} and C_{out} represent the input and output channels of the convolution layers, respectively.

Standard 3D Convolution. Given an input video tensor of size $T \times H \times W$, a standard 3D convolution with a cubic kernel size of $N \times N \times N$ incurs a computational cost (measured in FLOPs) of:

$$\text{Cost}_{\text{std}} = T \cdot H \cdot W \cdot C_{\text{in}} \cdot C_{\text{out}} \cdot N^3. \tag{7}$$

Depthwise Separable 3D Convolution. The proposed mechanism splits the computation into two steps:

1. **Depthwise Convolution:** We first apply spatiotemporal filters of size $N \times N \times N$ to each input channel independently. This step captures both motion and spatial patterns while maintaining channel independence, with a computational cost (in FLOPs) of:
$$\text{Cost}_{\text{dw}} = T \cdot H \cdot W \cdot C_{\text{in}} \cdot N^3. \tag{8}$$
2. **Pointwise Convolution:** We then apply a $1 \times 1 \times 1$ convolution to linearly combine the filtered features across channels, with a computational cost (in FLOPs) of:
$$\text{Cost}_{\text{pw}} = T \cdot H \cdot W \cdot C_{\text{in}} \cdot C_{\text{out}}. \tag{9}$$

Efficiency Analysis. The total cost of the depthwise separable 3D convolution, denoted by Cost_{sep}, is the sum of Cost_{dw} and Cost_{pw}, i.e., $\text{Cost}_{\text{sep}} = \text{Cost}_{\text{dw}} + \text{Cost}_{\text{pw}}$. The reduction ratio of the computational cost relative to standard 3D convolution is derived as:

$$\frac{\text{Cost}_{\text{sep}}}{\text{Cost}_{\text{std}}} = \frac{T \cdot H \cdot W \cdot C_{\text{in}}(N^3 + C_{\text{out}})}{T \cdot H \cdot W \cdot C_{\text{in}} \cdot C_{\text{out}} \cdot N^3} = \frac{1}{C_{\text{out}}} + \frac{1}{N^3}. \tag{10}$$

For a typical kernel size where $N = 3$, the term $\frac{1}{N^3}$ is equal to $\frac{1}{27}$. As C_{out} increases, $\frac{1}{C_{\text{out}}}$ becomes negligible. Consequently, this design reduces the computational load and parameter count by approximately 27 times compared to standard 3D convolutions.

A.2. Detailed Analysis of the Effectiveness of Shortcut Injection

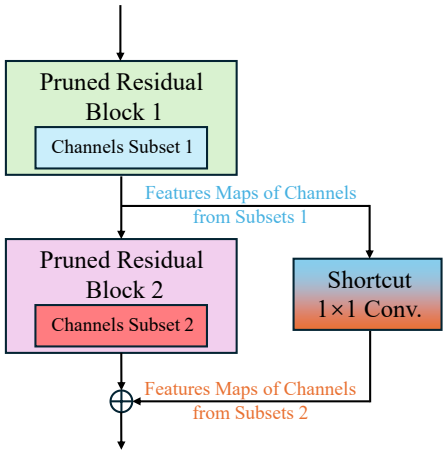


Figure 12. **Effectiveness of shortcut injection.**

In this section, we detail why retaining different indices between residual blocks destroys the continuity of the original model. We also explain how our pruning strategy and 1×1 convolutional shortcut injection effectively address this issue.

A standard residual block consists of two parallel pathways: the main path, which comprises stacked convolutional layers that transform the features, and the shortcut path, which directly links the input to the output to prevent gradient vanishing.

Typically, an identity mapping is employed for the shortcut path. However, when the retained channel indices of two consecutive blocks are different, using an identity shortcut become problematic. Specifically, it results in the addition of feature maps from Channel Subset 1 (the output of Residual Block 1) to feature maps from Channel Subset 2 (the output of Residual Block 2), thereby destructing the continuity of the original model. To address this, our pruning strategy ensures that the feature maps from the retained channels of Residual Block 1 can linearly represent the feature map from any channel within the same

block. Consequently, we can establish a mapping from these retained channels to the specific channels in Residual Block 1 that correspond to the retained indices of Residual Block 2. This mapping matrix, denoted by $\mathbf{W}$, is computed using the least squares method and is then used to initialize an injected 1×1 convolutional shortcut. This approach aligns the channel features from the shortcut pathway with those from the main path of the Pruned Residual Block 2, allowing for the effective summation of outputs and thus restoring connectivity within the model.

As for the main path, although no projection layer is injected, it does not suffer from topological discontinuity. Since the feature maps output from Residual Block 1 can linearly reconstruct the feature map from any channel, the subsequent convolutional layers in Residual Block 2 can readily learn the mapping from the input to the desired feature representation.

A.3. Gradient Mask for Training Stage 2

In the second training stage, i.e., pre-pruning retained channel enhancement, we employ a gradient mask to prevent all channels from converging towards the retained channels. Specifically, for the to-be-pruned channels, we suspend weight updates, preserving their original information while compelling the retained channels to fit the full-channel representation.

A.4. Additional Details for the Visual Comparison of Video Generation Results

Due to space constraints in the main text, we present only partial prompts. Here, we provide the complete prompts used for visual comparisons.

For the visual comparison of Self Forcing, the full prompt employed is: *A stylish woman strolls down a bustling Tokyo street, the warm glow of neon lights and animated city signs casting vibrant reflections. She wears a sleek black leather jacket paired with a flowing red dress and black boots, her black purse slung over her shoulder. Sunglasses perched on her nose and a bold red lipstick add to her confident, casual demeanor. The street is damp and reflective, creating a mirror-like effect that enhances the colorful lights and shadows. Pedestrians move about, adding to the lively atmosphere. The scene is captured in a dynamic medium shot with the woman walking slightly to one side, highlighting her graceful strides.*

For FastVideo, the full prompt employed is: *A fluffy golden retriever sits calmly in a sunlit backyard, surrounded by vibrant green grass and colorful flower beds, its eyes alert and ears perked. Suddenly, with a burst of energy, it leaps up and begins to run in joyful circles, tail wagging furiously. The camera captures its playful antics, zooming in on its gleeful expression and the way its fur catches the sunlight. As it races around, the flowers sway gently in the breeze, and the sound of its paws thumping on the ground adds to the lively atmosphere. Finally, it pauses, panting happily, before flopping down on the grass, basking in the warmth of the day.*

Additionally, for the FastVideo visual comparison, we replace the VAE decoder in the Fastvideo pipeline with the original VAE decoder from Wan 2.1 to ensure fairness. Note that Fastvideo recommends an inference resolution of 448×832, and its VAE decoder is specifically fine-tuned for this resolution. In contrast, both our VAE decoder and the Wan 2.1 VAE decoder are trained at a resolution of 480×832. Therefore, to maintain a fair comparison at the same resolution, we use the original Wan 2.1 VAE decoder. However, it is important to note that this replacement is not applied in quantitative evaluations.

B. Supplementary Experimental Results

B.1. Ablation on Resolution and Number of Frames.

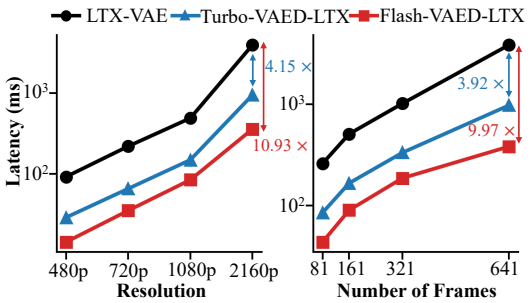


Figure 13. Ablation results of using different resolutions and numbers of frames.

To evaluate the performance of Flash-VAED in computation-intensive scenarios, we systematically increase the resolution and the number of frames in the decoded videos, and plot the latency variations in Figure 13. It is evident that as the resolution and number of frames increase, the latency of Flash-VAED scales more slowly than that of both the original VAE decoders and Turbo-VAED. Consequently, under conditions of high resolution and large frame count, our method achieves a significant acceleration ratio, offering approximately a $10\times$ speedup. The experiments confirm that Flash-VAED exhibits superior scalability, which unleashes its potential in high-compute scenarios.

B.2. Latency Breakdown of Generation Pipelines

To further verify the latency bottleneck shift induced by the acceleration of the DiT, we conduct a component-wise latency breakdown of the full generation pipeline. We evaluate the original Wan 1.3B model (Wan et al., 2025) alongside the Self Forcing (Huang et al., 2025a) and FastVideo (Zhang et al., 2025) optimized variants, as detailed in Table 5. The results indicate that text encoding contributes only a negligible portion of the total latency in both the baseline and accelerated pipelines. In contrast, the relative latency associated with the decoding stage becomes significantly more pronounced, with its percentage contribution increasing by approximately 15× following the DiT acceleration.

Table 5. Latency breakdown of generation pipelines

Model	Text Encoding	Denoising	Video Decoding	Decoding Percentage
Wan 1.3B (Wan et al., 2025)	0.619	178.066	4.212	2.3%
Self Forcing-Wan 1.3B (Huang et al., 2025a)	0.598	6.254	3.172	31.6%
FastVideo-Wan 1.3B (Zhang et al., 2025)	0.539	5.136	3.925	40.9%

B.3. Comparison of SOTA video VAE decoders

Table 6. To validate our choices of Wan (Wan et al., 2025) and LTX-Video (HaCohen et al., 2024) as original VAE decoders, we compare them against current SOTA video VAE decoders. Wan establishes a clear lead in both quality and speed among models with mainstream compression ratios, while LTX-Video dominates in terms of inference speed.

Model	$(\mathrm{d_T}, \mathrm{d_H}, \mathrm{d_W})$	**FPS ↑**	**Reconstruction Quality**		
		RTX 5090D	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**
Wan 2.1 (Wan et al., 2025)	$(4, 8, 8)$	19.27	40.40	0.9733	0.0190
HunyuanVideo (Kong et al., 2024)	$(4, 8, 8)$	6.77	33.05	0.9392	0.0272
CogVideoX (Yang et al., 2024)	$(4, 8, 8)$	4.99	32.81	0.9351	0.0302
LTX-Video (HaCohen et al., 2024)	$(8, 32, 32)$	204.55	33.28	0.9253	0.0497