

# REBUTTAL FOR UNIVAE

**Anonymous authors**

Paper under double-blind review

We thank all the reviewers for their efforts and constructive comments on this paper. Here, we will provide detailed responses to each question raised by each reviewer.

## 1 TO REVIEWER DX2P

### 1.1 CLARIFICATION ON OUR MOTIVATION AND TECHNICAL CONTRIBUTION.

We sincerely appreciate your insightful comments and will revise our wording and discussion accordingly. In our humble opinion, joint spatiotemporal modeling using multi-scale convolution in video VAE is a reasonable and intriguing approach for both diffusion and interpolation, which may be beneficial to share with the field timely.

**Q1: The introduction does not convincingly explain why we need the proposed multi-kernel convolution approach.**

**A1:** Thanks. The challenges of handling video data largely come from their multi-resolution nature in both spatial and temporal domain, which has been the research theme for multidecade. The multi-scale convolution with a large receptive field in VAE offers one way to enhance compression and reconstruction quality of *long-duration* and *high frame rate* videos, especially using joint spatiotemporal modeling at the feature level in one framework for both diffusion and interpolation tasks. We would not claim multi-scale convolution is the only way, yet we claim it deserves the research efforts for its potential in VAE and video regeneration.

From the perspective of experiments results:

(1) Single-scale convolution kernel results in video reconstruction performance degradation. As illustrated in Fig. 1, both OS-VAE and OD-VAE which use single-scale convolution kernel exhibit performance degradation when reconstructing long videos, due to the limited ability of single-scale convolution kernel to capture changes occurring over varying time scales in videos.

(2) Single-scale convolution cannot support our refinement decoder design. Compared with single-scale convolution, multi-scale convolution kernel can better extract temporal patterns at various time scales, which can provide the refinement decoder more informative features for additional intermediate frame generation. Our multi-scale convolution kernels are indispensable for the subsequent refinement decoder. To demonstrate the advantage of multi-scale convolution kernels for additional frame generation, we train a new refinement decoder based on pre-trained OD-VAE that only uses single-scale convolution kernel when temporal downsampling. The comparison in Fig. 2 shows UniVAE with multi-scale convolution kernels can generate better intermediate frames compared with OD-VAE with single-scale convolution.

From the perspective motivation:

We aim to improve VAE performance by enhancing the encoder’s ability to capture temporal clues for spatio-temporal modeling. Our multi-scale temporal convolution kernels endow the UniVAE ability to perceive and capture the dynamics patterns across different time scales in videos, which can result in better video reconstruction and generation results compared with existing video VAE, as shown in Tab. 1 and Tab. 2.

**Q2: Why does temporal pooling operation limit the ability of video VAEs (Line 51)?**

The temporal pooling operation is a rudimentary approach to temporal dimensionality reduction, as evidenced by various improved methods [Ruderman et al. \(2018\)](#) developed for different tasks [Kauderer-Abrams \(2017\)](#); [Long et al. \(2015\)](#) over time. Fixed pooling operations may miss visual or motion details and lack of learnability. Instead, 3D convolutions generally are robust to

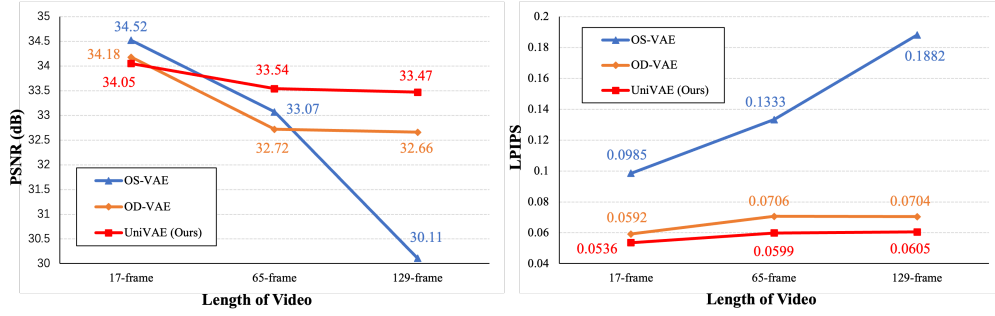


Figure 1: Performance (PSNR and LPIPS) comparison of different VAEs on video reconstruction across different frames.

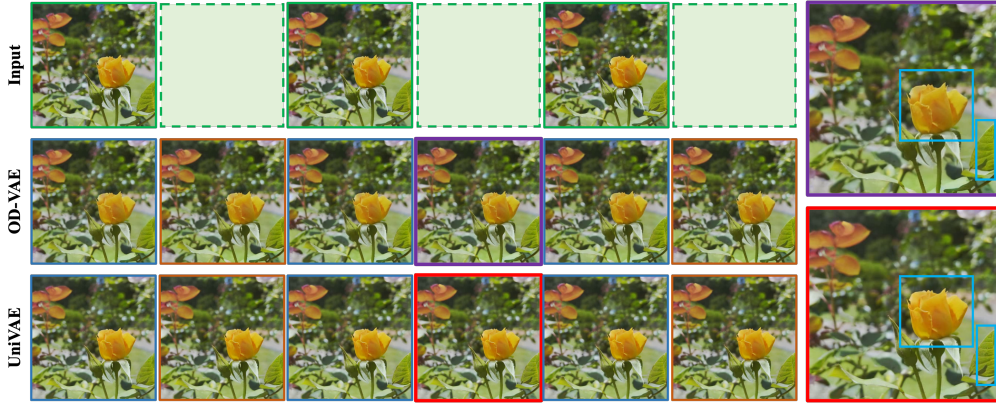


Figure 2: Qualitative comparison of intermediate frames generated by UniVAE and OD-VAE.

capture more appearance details and varying motion patterns [Yu et al. \(2023\)](#). To further support this argument, we replace the temporal downsampling operator in our UniVAE with average pooling, and shows the comparison results in Tab. 3. As we can see, our UniVAE achieves better performance than VAE with temporal pooling operator.

**Q3: Why would applying different kernels across channel partitions necessarily facilitate better temporal modeling? How did the authors decide to apply the kernels for each channel partition? Which partition uses the smaller temporal kernels and the larger ones?**

**A3:** (1) The multi-scale convolution kernels provide varying receptive fields, enabling the model to capture temporal modeling across different scales. This statement is supported by evidence from [Wang et al. \(2018\)](#). Moreover, our ablation study in Tab. 4 also supports this point.

(2) The application of channel partitions is to reduce the computation cost. In fact, we can re-design several new convolution kernels  $\mathbf{F} = \{f_1^{new}, f_2^{new}, \dots, f_p^{new}\} \in \mathbb{R}^{C \times C_{out} \times t \times h \times w}$ , and directly apply them to the feature  $\mathbf{x} \in \mathbb{R}^{N \times H \times W \times C}$  without channel partition, then fuse them together and get the final result  $\mathbf{y} = fusion(f_1^{new} \otimes \mathbf{x}, f_2^{new} \otimes \mathbf{x}, \dots, f_p^{new} \otimes \mathbf{x})$ . However, compared with single-scale convolution, this design introduces  $(p - 1)$  additional convolution kernels, increasing the computational load. To make a better trade-off between effectiveness and efficiency, we draw inspiration from [Ding et al. \(2024\)](#) and choose to apply different convolution kernels across channel partitions as described in Sec.3.2. The multi-scale convolution kernels can encourage VAE to capture the dynamic patterns across different time scales, while the application of channel partition reduces the parameters of each convolution kernel, to lower the overall computational cost of VAE.

(3) Detailed application: As detailed in Sec.3.2, we first evenly split the input  $\mathbf{x}$  along the channel dimension into  $p$  parts, which is  $\mathbf{x} = [x_1, x_2, \dots, x_p]$ . Then, we arrange the multiply convolutions in ascending order of kernel size and get the  $\mathbf{F} = \{f_1, f_2, \dots, f_p\}$ . After that, we perform convolution between each kernel  $f_i$  and segment  $x_i$ , and concatenate the result along the channel dimension to get the final output, which is:

Table 1: Performance comparison of different VAEs on 25-frame video reconstruction across WebVid-10M and Panda-70M dataset. The best results are marked as **bold**, and the second ones are marked by underline. Note that the **UniVAE** \* denotes that the UniVAE w/o refinement decoder  $\mathcal{D}_2$ , since only  $\mathcal{D}_1$  are utilized for standard video reconstruction.

Method	VCR	Params	WebVid-10M			Panda-70M		
			PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
VQGAN	1 $\times$ 8 $\times$ 8	69.0M	26.26	0.7699	0.0906	26.07	0.8295	0.0722
SD-VAE	1 $\times$ 8 $\times$ 8	83.7M	30.19	0.8379	0.0568	30.40	0.8894	0.0396
SVD-VAE	1 $\times$ 8 $\times$ 8	97.7M	31.15	0.8686	0.0547	31.00	0.9058	0.0379
TATS	4 $\times$ 8 $\times$ 8	52.2M	23.10	0.6758	0.2645	21.77	0.6680	0.2858
CV-VAE	4 $\times$ 8 $\times$ 8	182.5M	30.76	0.8566	0.0803	29.57	0.8795	0.0673
OS-VAE	4 $\times$ 8 $\times$ 8	393.3M	31.12	0.8569	0.1003	<u>31.06</u>	0.8969	0.0666
OD-VAE	4 $\times$ 8 $\times$ 8	239.2M	<u>31.16</u>	0.8694	0.0586	30.49	0.8970	0.0454
<b>UniVAE *</b>	4 $\times$ 8 $\times$ 8	234.8M	<b>34.13</b>	<b>0.8783</b>	<b>0.0525</b>	<b>33.58</b>	<b>0.9138</b>	<b>0.0444</b>

Table 2: Performance comparison of different VAEs on video generation across UCF101 and Sky-Timelapse dataset. The best results are marked as **bold** and the seconds one are marked by underline.

Method	UCF101		SkyTimelapse	
	FVD $\downarrow$	KVD $\downarrow$	FVD $\downarrow$	KVD $\downarrow$
Latte + CV-VAE	8742.42	20.13	986.30	11.25
Latte + OD-VAE	8047.60	20.65	881.66	<b>9.41</b>
Latte + UniVAE	<b>7777.71</b>	<b>19.35</b>	<b>799.64</b>	<u>9.56</u>

Table 3: Performance comparison between different VAEs on video reconstruction.

Method	17-frame		65-frame		129-frame	
	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
Pooling-VAE	33.91	0.0574	33.48	0.0621	33.41	0.0623
UniVAE	34.05	0.0536	33.54	0.0599	33.47	0.0605

$$\mathbf{y} = \mathbf{F} \otimes \mathbf{x} = [f_1 \otimes x_1, f_2 \otimes x_2, \dots, f_p \otimes x_p].$$

Table 4: Ablation results of UniVAE for video reconstruction on WebVid-10M validation set.

Method	Settings of $\mathbf{F}$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Baseline	$\mathbf{F} = [3]$	31.16	0.8694	0.0586
UniVAE-V	$\mathbf{F} = [3, 5]$	34.08	<b>0.8785</b>	0.0527
UniVAE	$\mathbf{F} = [3, 5, 7, 9]$	<b>34.13</b>	0.8783	<b>0.0525</b>

**Q4:** Why is the multi-kernel convolution applied only during downsampling? Did the authors employ the same scheme in the upsampling stage? As the spatiotemporal dimension has to match across the partitions, a larger temporal kernel also implies using larger temporal padding and stride which could be counterintuitive to the basic benefit the authors are arguing.

**A4:** We do not employ the multi-kernel convolution in the upsampling since the enhanced encoded latent representations are sufficient to reconstruct a video with simple upsampling. Below, we provide a detailed explanation along with supplementary experimental evidence.

In VAE, the role of encoder  $\mathcal{E}$  is to map input videos to the latent representation  $\mathbf{Z}$ . When performing temporal compression, the downsampling operation inevitably leads to information loss. To address this, we introduce multi-scale kernels for temporal downsampling. This allows  $\mathcal{E}$  to capture temporal dynamics at multiple time scales, reducing information loss caused by single-scale downsampling and improving the quality of the latent representation. Compared to the decoder, the encoder plays a more critical role, as it compresses the video into the latent representation that will be utilized to train the latent video diffusion models. If too much information is lost during downsampling, the decoder will struggle to reconstruct high-quality video.

On the other hand, decoder is responsible for reconstructing input videos from the latent representation  $\mathbf{Z}$  produced by  $\mathcal{E}$ . Although using multi-scale kernels in the decoder may enhance the reconstruction quality, our primary focus is on the encoder, as the quality of the encoded latent representation directly impacts the subsequent diffusion models. Therefore, we prioritize applying multi-scale kernels in encoder.

We further design a variant that adopt multi-scale convolution kernels on both downsampling and upsampling modules, which is denoted as “UniVAE-V3”. In Tab. 5, the results show that given the improved latent space obtained by multi-scale convolutions, simple upsampling can do a fairly good job to reconstruct the video in terms of PSNR and LPIPS.

Table 5: Performance comparison between different VAEs on video reconstruction.

Method	17-frame		65-frame		129-frame	
	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
UniVAE-V3	33.68	0.0614	32.98	0.0731	32.88	0.0753
UniVAE	34.05	0.0536	33.54	0.0599	33.47	0.0605

**Q5: If the interpolation decoder is trained independently, isn’t it the same as training a separate interpolation model (given that the encoder and the reconstruction decoder are frozen)?**

**A5:** We do not agree the training of the refinement decoder is the same as training a separate interpolation model. Similar with Imagen Saharia et al. (2022) that utilizes super-resolution diffusion models to upsample the generated image, the refinement decoder is a component of the UniVAE. There are two key difference between the refinement decoder and a standalone frame interpolation method. (a) The frame interpolation methods typically can only generate the intermediate frame  $x_{new}$  based on the given frames  $[x_1, x_2]$ . In contrast, our refinement decoder can directly reconstruct the video and the generated additional frame  $[x_1, x_{new}, x_2]$  in a single pass. (b) The frame interpolation methods are typically independent and do not rely on video generation techniques. In contrast, our refinement decoder generate video and additional frames based on the latent features extracted by the VAE encoder, which is integrated with our UniVAE.

**Q6: What is the benefit of adding the reconstructed frames in the interpolation decoder?**

**A6:** As described in Sec.3.3, adding the reconstructed frames ( $\mathcal{D}_1$ ) in the interpolation decoder ( $\mathcal{D}_2$ ), reducing  $\mathcal{D}_2$ ’s burden of reconstructing existing frames and allowing  $\mathcal{D}_2$ ’s to focus more on generating additional frames (line 264-267). Also, we demonstrate the effectiveness of this design in Fig.7 in ablation study. As we can see, our latent-guided refinement training strategy can better preserve hand details (specific in Fig. 4).

**Q7: How does the overall approach compare with just using a pre-trained, state-of-the-art interpolation model on top of the reconstructed frames? There should be a more thorough analysis in this regard.**

**A7:** Here, we provide a thorough analysis with existing frame interpolation method.

(1) **Performance comparison:** We choose FILM Reda et al. (2022) as baseline, and compare it with our UniVAE on frame interpolation. FILM is an independent frame interpolation method, which designs flow estimation modules to compute flows based on feature pyramids. For FILM, we first send the input video into the UniVAE and get the ordinary output from the regular decoder  $\mathcal{D}_1$ , and then use pre-trained FILM for frame interpolation. The input videos are pre-processed to a length of 65 frames with a resolution of  $512 \times 512$ . We show the qualitative results in Fig. 3. As we can see, our UniVAE achieve comparable interpolation results with the specifically designed interpolation method FILM.

(2) **Time consumption:** We further provide the time consumption for the two pipelines, *i.e.*, “UniVAE” and “VAE + FILM”. As described in (1), for each 65-frame video with a resolution of  $512 \times 512$ , we apply both interpolation methods to extend it to 129 frames, respectively. We show the time consumption of these two methods in Tab. 6. Among them, “UniVAE” denotes we directly leverage the UniVAE to perform interpolation. The “VAE” and “FILM” mean UniVAE reconstruction and subsequent separate frame interpolation, respectively. As we can see, our UniVAE is more efficient than “VAE + FILM”, which further prove the potential of UniVAE in supporting latent video diffusion models.



(3) **More discussion:** As the first attempt to utilize VAE for video interpolation, we focus on exploring the possibility and potential of video VAE for more frame generation. Compared to traditional independent interpolation methods, our UniVAE offers better coherence and integration. Moreover, the experiment results in (1) and (2) show that even without dedicated interpolation modules (such as flow estimation modules in FILM), our UniVAE still achieve comparable performance to existing interpolation methods while reducing complexity of pipeline and improving efficiency. This is useful for further advancing the latent video diffusion models.



Figure 3: Performance comparison between UniVAE and FILM on frame interpolation.

Table 6: Time consumption comparison between “UniVAE” and “VAE + FILM”.

Method	UniVAE	VAE + FILM		
		VAE	FILM	Total
Time Consumption	22.83s	6.72s	464s	470.72s

**Q8:** (1) The claim that this is the first paper to unify spatial and temporal modeling in the encoder is too strong and not correct (Line 111). Many previous works such as OS-VAE and OD-VAE do both spatial and temporal compression. (2) The authors claim their work is specifically for video data (Line 106). Then, what is the motivation for using causal 3D convolutional networks which are commonly used for joint image and video encoding/decoding? Why formulate the problem using  $N+1$  frames?

**A8:** We apologize for our unclear and confusing wording. Here, we clarify them.

(1) **The claim in Line 111:** Spatiotemporal compression does not equate to unify spatial and temporal modeling, as there are various methods to achieve spatiotemporal compression, such as “tandem” VAE (*e.g.*, OS-VAE), OD-VAE uses 3D convolution for spatial and temporal modeling. We will revise it and update the sentence to “multi-scale spatiotemporal modeling in the encoder”.

(2) **The claim in Line 106:** Causal 3D Convolutional Networks are widely utilized in video VAEs, which allows them to encode/decode image and video jointly. However, in this paper, we focus on their capability in videos, aligning with the motivations of some video VAE works like Yang et al. (2024); Zheng et al. (2024); Chen et al. (2024). Additionally, the “ $N+1$  frames” is a common practice in VAEs, particularly in video diffusion models, and we have adopted it in our work. We will revise the expression to make it clearer.

## 1.2 CLARIFICATION ON EXPERIMENTAL DETAILS.

**More training steps in stage 2.** In stage 2, we aim to train the refinement decoder to estimate the additional intermediate frames based on the latent feature  $Z$ , which is more difficult than the reconstruction in stage 1. So we train the refinement decoder for more steps in stage 2.

**The coefficients used for different loss functions.** Following the common setting in previous video VAE training Chen et al. (2024), we set the coefficients of reconstruction loss, adversarial loss, and KL regularization as 1, 5000, and  $1e-06$ , respectively.

**GAN training.** The GAN training starts after 2000 steps in stage 1.

**The loss functions in stage 1 and stage 2.** Both stage 1 and stage 2 use the same set of loss functions in Eq.1. While, the encoder  $\mathcal{E}$  and the regular decoder  $\mathcal{D}_1$  are frozen in this process.

**The  $256 \times 256$  resolution video for evaluation.** For fair comparison, we follow the common settings in previous works [Chen et al. \(2024\)](#); [Zheng et al. \(2024\)](#), which typically transform videos to clips of 25-frame length and  $256 \times 256$  resolution for evaluation.

**Training dataset and Model Initialization.** (1) Training dataset: We collect 1.5M videos from Internet, and use them to train our UniVAE. Since most of the data for training VAEs is private and unavailable [Zheng et al. \(2024\)](#); [Chen et al. \(2024\)](#), it is difficult to set them uniformly. (2) Model Initialization: For training efficiency, we initialize part of modules in our UniVAE with parameters from OD-VAE to provide UniVAE with an initial video encoding and decoding capability, while the other part of modules are initialized randomly.

**Clarification on video generation performance and details.** The high FVD values in Tab.2 are due to the insufficient training of the latent diffusion model (Latte), which is not counterintuitive to the reconstruction performance of VAEs. When training the latent diffusion model, we train Latte equipped with different VAEs on UCF101 and SkyTimelapse for 100K steps, due to the time and computational constraints. Each training sample is pre-processed to a length of 24 frames with a resolution of  $256 \times 256$ . During evaluation, we generate 100 samples with 24-frame length and  $256 \times 256$  resolution to calculate the FVD and KVD metrics.

### 1.3 CLARIFICATION ON EXPERIMENT AND ABLATION RESULTS.

**Q1: The authors fail to convincingly argue why their proposed approach gives a very strong performance compared to previous methods in Section 4.2. What makes the proposed approach very unique to achieve such a performance? Does using multiple kernels give such a performance boost?**

**A1:** The performance improvement is attributed to our multi-scale convolution kernels in temporal downsampling module. Compared to other methods [Zhao et al. \(2024\)](#); [Zheng et al. \(2024\)](#); [Chen et al. \(2024\)](#), we introduce multi-scale convolution kernels for temporal compression to our UniVAE, which endows it with better capability to capture dynamic patterns of different time-scales in videos. As a result, our UniVAE achieves better reconstruction and generation performance as shown in Tab. 1 and Tab. 2.

**Q2: Details about experiment in Sec.4.4.**

**A2:** (1) Dataset: The 1000 videos used in Sec.4.4 for evaluation are selected from WebVid-10M dataset. (2) Performance: When reconstructing 17-frame videos, all three methods can reconstruct video accurately. That’s reason why OS-VAE, OD-VAE, and UniVAE achieve comparable performance. On the other hand, for efficiency, we select a small subset from WebVid-10M for evaluation in Sec.4.4 (as described in (1)), which leads to some differences compared to the results in Tab.1, where the whole WebVid-10M dataset is used.

**Q3: The Fig.7 in ablation study.**

**A3:** Here, present detailed comparison of the fourth images of Fig.7 in Fig. 4, which are the additional intermediate frames generated by UniVAE-V2 and UniVAE, respectively. As we can see, the UniVAE can better preserve hand detail features than UniVAE-V2.

### 1.4 THE QUALITATIVE ANALYSIS IS LIMITED.

Thanks for your suggestion. We have added more qualitative results in the updated appendix to provide clearer visual comparison. For video reconstruction, we put qualitative results in “*.zip/Qualitative\_Results/Reconstruction/Reconstruction\_1.mp4*” and “*Reconstruction\_2.mp4*”. For video generation, we put them in “*.zip/Qualitative\_Results/Generation/Generation.mp4*”.

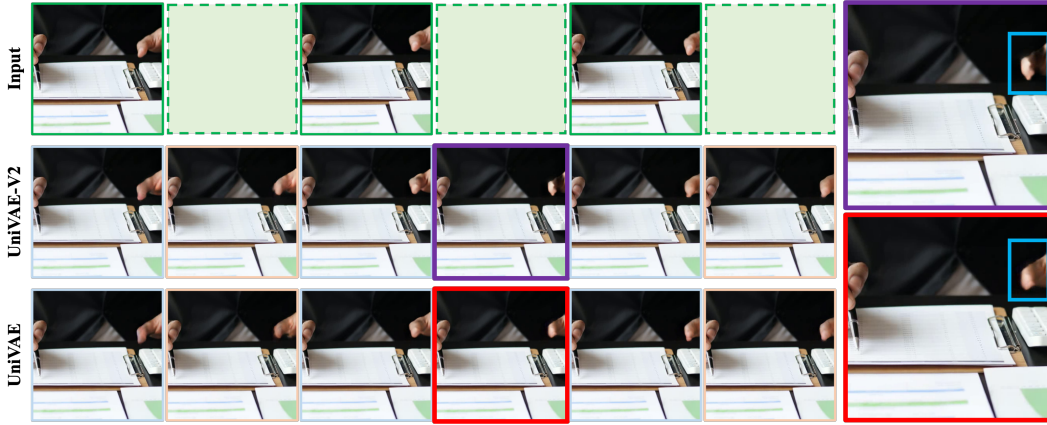


Figure 4: Qualitative comparison of the UniVAE and UniVAE-V2.

## 2 TO REVIEWER NOHP

**Q1: The effectiveness of the refinement decoder is not sufficiently demonstrated, as it relies solely on the qualitative analysis presented in Figure 5. However, I have observed a color bleeding phenomenon in the interpolated images. While this is a good idea, I would appreciate more discussion on this aspect.**

**A1:** (1) As the original work to leverage VAE for generating additional frames, beyond existing video VAE, we focus on exploring the possibility and potential of using VAE decoder for frame interpolation. Thus, unlike current frame interpolation techniques, we do not design specific modules (like motion prediction module) in the refinement decoder, which may lead to detailed issues such as color bleeding in the generated intermediate frames. However, as shown in Fig. 7, even without any specially designed motion estimation modules, our UniVAE can still accurately predict the motion trajectories in videos, and generate precise intermediate frames.

(2) To further prove the effectiveness of our UniVAE for frame interpolation, we compare our UniVAE with existing separate interpolation method FILM [Reda et al. \(2022\)](#). FILM is an independent frame interpolation method, which designs flow estimation modules to compute flows based on feature pyramids. For FILM, we first send the input video into the UniVAE and get the ordinary output from the regular decoder  $\mathcal{D}_1$ , and then use pre-trained FILM for frame interpolation. We denote it as “VAE + FILM”.

(a) Performance comparison: The input videos are pre-processed to a length of 65 frames with a resolution of  $512 \times 512$ . We show the qualitative results in Fig. 5. As we can see, our UniVAE achieve comparable interpolation results with the specifically designed interpolation method FILM.

(b) Time consumption: We further provide the time consumption for the two pipelines, *i.e.*, “UniVAE” and “VAE + FILM”. As described in (1), for each 65-frame video with a resolution of  $512 \times 512$ , we apply the both interpolation methods to extend it to 129 frames, respectively. We show the time consumption of these two methods in Tab. 7. As we can see, video reconstruction is faster than video interpolation in UniVAE. However, the subsequent separate interpolation will cost more time than UniVAE. As a result, our UniVAE is more efficient than “VAE + FILM”, which further prove the potential of UniVAE in supporting latent video diffusion models.

The experiment results in Fig. 5 and Tab. 7 show that even without dedicated interpolation modules (such as flow estimation modules in FILM), our UniVAE still achieve comparable performance to exiting interpolation methods while reducing complexity of pipeline and improving efficiency. This is crucial for further advancing the latent video diffusion models.



Figure 5: Performance comparison between UniVAE and FILM on frame interpolation.

Table 7: Time consumption comparison between “UniVAE” and “VAE + FILM”.

Method	UniVAE	VAE + FILM		
		VAE	FILM	Total
Time Consumption	22.83s	6.72s	464s	470.72s

**Q2:** Given that most current VAEs utilize tiling to reduce inference memory, I wonder if the Multi-Scale Spatial-Temporal downsampling method imposes any limitations within tiling. Can it still ensure satisfactory performance in such scenarios?

**A2:** Our spatial-temporal downsampling imposes no limitations within tiling. In fact, following the previous methods [Chen et al. \(2024\)](#), we also employ tiling technique when evaluating the performance of our UniVAE, and show the results in Tab. 1, Tab. 2, and Fig. 1.

**Q3:** Have the authors attempted to use latent videos generated by diffusion models (e.g., Latte) as input to the refinement decoder? This approach may pose greater challenges compared to directly encoding videos and performing interpolation, but it could also provide stronger evidence for the significance of the proposed method.

**A3:** Thanks for your good suggestion. Since the regular decoder  $\mathcal{D}_1$  and the refinement decoder  $\mathcal{D}_2$  share the latent representation  $\mathbf{Z}$ , the Latte equipped with our UniVAE can directly leverage the refinement decoder to output videos with richer content. We show some qualitative results in Fig. 6.

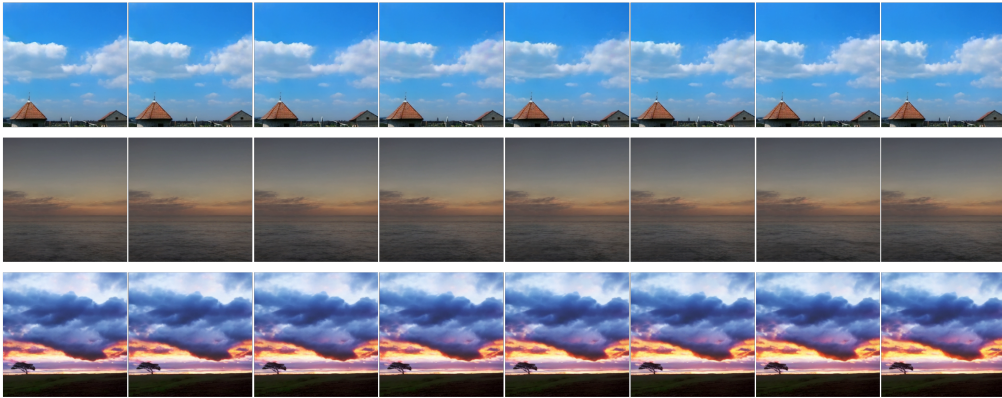


Figure 6: Qualitative comparison of intermediate frames generated by UniVAE and OD-VAE.



### 3 TO REVIEWER KWYS

**Q1: The writing of the paper left me somewhat confused. In lines 79-80, what does “Our observations indicate...” refer to? How did the authors come to the subsequent conclusion that “If the encoder can effectively model temporal variations, the decoder could theoretically synthesize frames, leading to high fps videos without the for separate interpolation models”?**

**A1:** We apologize for the confusing expression in lines 79-80.

(1) “Our observations” refer to the experiment results in Sec.4.2 and Sec.4.3. As we can see, our UniVAE not only fulfills the typical VAE role of video encoding and decoding in latent video diffusion models (Tab. 1 and Tab. 2), but also extends to generate additional intermediate frames, as shown in Fig. 7. This is what we refer to in lines 79-80 as “VAEs can assume multiple roles in current video generation stream.”

(2) If the encoder of the VAE effectively captures the temporal dynamics of a video, the latent features  $\mathbf{Z}$  will contain rich information about the sequence changes, such as object motion trajectories. This allows the decoder to leverage these dynamic patterns in  $\mathbf{Z}$  to not only reconstruct the original video frames but also generate intermediate frames between them, leading to high fps videos without separate interpolation models. As we can see in Fig. 7, the refinement decoder can leverage the rich temporal information in  $\mathbf{Z}$  to predict the motion trajectories and generate precise intermediate frames.

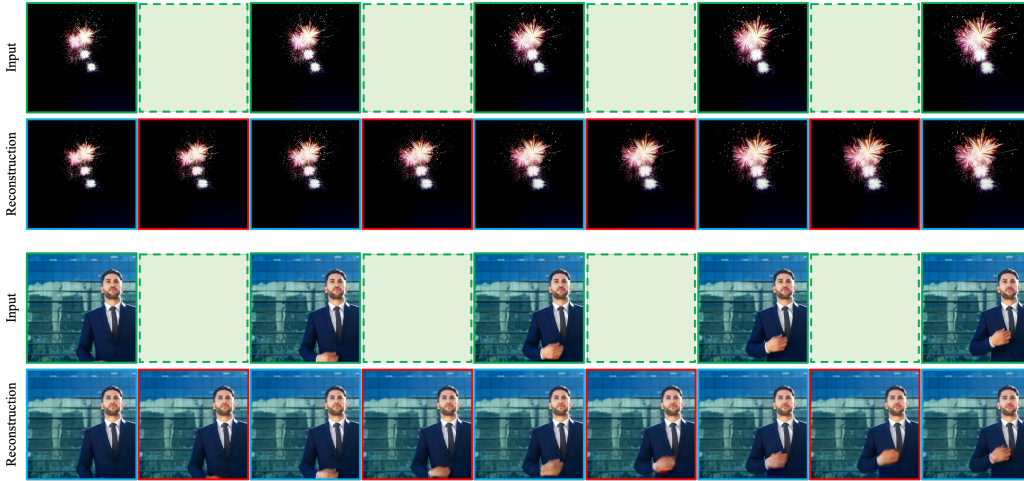


Figure 7: Reconstruction results of the refinement decoder  $\mathcal{D}_2$ .

**Q2: The paper emphasizes that both the Encoder and Decoder in UniVAE are causal. What specifically does “causal” mean in this context? What would be the results if either or both were non-causal?**

**A2:** The term “causal” indicates that the VAE processes the relationships between frames in a causal manner when encoding and decoding video. Specifically, when handling frame  $f_i$ , it only considers the current frame  $f_i$  and the preceding frames  $[f_1, f_2, \dots, f_{i-1}]$ , without referencing subsequent frames, which means the output for each frame only depends on the previous frames. This design enables VAE to encode both video and image Yu et al. (2023). The causal setup is widely applied in decoder-only architectures of large language models (LLMs) and video VAEs Zheng et al. (2024); Zhao et al. (2024); Chen et al. (2024).

**Q3: Could the latent-guided refinement training scheme proposed in this paper also be used to enhance the performance of other pre-trained VAE?**

**A3:** Yes. The latent-guided refinement training scheme can be utilized in other pre-trained video VAE. Here, we retrain OD-VAE with our proposed latent-guided refinement training scheme, and show the qualitative comparison in Fig. 8. As we can see, our latent-guided can also be utilized to enhance the output of OD-VAE. While, our UniVAE still obtains better results.



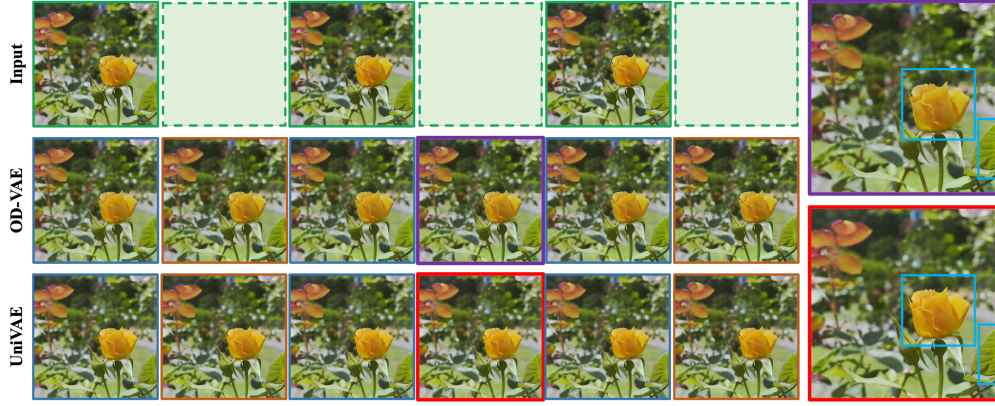


Figure 8: Qualitative comparison of intermediate frames generated by UniVAE and OD-VAE.

#### 4 TO REVIEWER UPGC

##### Q1: Clarification on the novelty of the proposed UniVAE.

**A1:** This is an original effort to unify video reconstruction and frame interpolation in one VAE framework for latent video generative models, not on the spatial and temporal compression. There are two key innovations compared to existing video VAEs.

(1) Multi-scale convolution kernels for temporal downsampling. We analyze the issues in existing video VAEs, and propose multi-scale temporal convolution kernels for temporal downsampling, which enhance the VAE’s ability to capture dynamic patterns in videos, as described in Sec.3.2. It brings better video reconstruction and generation results, as shown in Tab. 1 and Tab. 2.

(2) Frame interpolation within the UniVAE decoder. To the best of our knowledge, UniVAE is the first attempt to explore the potential of video VAEs to generate richer video content beyond reconstruction. We design the latent-guided refinement training strategy to enable UniVAE generate additional intermediate frames, which further allows latent video diffusion models to generate high fps videos with improved efficiency.

**Q2: Interpolation Decoder Training: The interpolation decoder is trained on its own after freezing the basic VAE encoder and decoder, making it more like an independent interpolation model. It would be more meaningful if the basic VAE encoder and decoder and the interpolation decoder are trained together to improve the representation ability of latent space.**

**A2:** (1) We do not agree the training of the refinement decoder is the same as training a separate interpolation model. Similar with Imagen [Saharia et al. \(2022\)](#) that utilizes super-resolution diffusion models to upsample the generated image, the refinement decoder is a component of the UniVAE. There are two key difference between the refinement decoder and frame interpolation method. (a) The frame interpolation methods typically can only generate the intermediate frame  $x_{new}$  based on the given frames  $[x_1, x_2]$ . In contrast, our refinement decoder can directly reconstruct the video and the generated additional frame  $[x_1, x_{new}, x_2]$  in a single pass. (b) The frame interpolation methods are typically independent and do not rely on video generation techniques. While, our refinement decoder generate video and additional frames based on the latent features extracted by the VAE encoder, which is integrated with our UniVAE.

(2) If we train the UniVAE encoder  $\mathcal{E}$ , regular decoder  $\mathcal{D}_1$ , and refinement decoder  $\mathcal{D}_2$  jointly, it may obscure the overall training objective of UniVAE. The optimization objective for video reconstruction in  $\mathcal{D}_1$  and additional frame generation  $\mathcal{D}_2$  may conflict, leading to unstable training.

##### Q3: Comparison with existing frame interpolation method.

**A3:** Here, we provide a analysis about the comparison between our UniVAE and existing frame interpolation method.

(1) Performance comparison: We choose FILM Reda et al. (2022) as baseline, and compare it with our UniVAE on frame interpolation. FILM is an independent frame interpolation method, which designs flow estimation modules to compute flows based on feature pyramids. For FILM, we first send the input video into the UniVAE and get the ordinary output from the regular decoder  $\mathcal{D}_1$ , and then use pre-trained FILM for frame interpolation. The input videos are pre-processed to a length of 65 frames with a resolution of  $512 \times 512$ . We show the qualitative results in Fig. 9. As we can see, our UniVAE achieve comparable interpolation results with the specifically designed interpolation method FILM.

(2) Time consumption: We further provide the time consumption for the two pipelines, *i.e.*, “UniVAE” and “VAE + FILM”. As described in (1), for each 65-frame video with a resolution of  $512 \times 512$ , we apply both interpolation methods to extend it to 129 frames, respectively. We show the time consumption of these two methods in Tab. 8. Among them, “UniVAE” denotes we directly leverage the UniVAE to perform interpolation. The “VAE” and “FILM” mean UniVAE reconstruction and subsequent separate frame interpolation, respectively. As we can see, our UniVAE is more efficient than “VAE + FILM”, which further validate the potential of UniVAE in supporting latent video diffusion models.

(3) More discussion: As the first attempt to utilize VAE for video interpolation, we focus on exploring the possibility and potential of video VAE for more frame generation. Compared to traditional independent interpolation methods, our UniVAE offers better coherence and integration. Moreover, the experiment results in (1) and (2) show that even without dedicated interpolation modules (such as flow estimation modules in FILM), our UniVAE still achieve comparable performance to existing interpolation methods while reducing complexity of pipeline and improving efficiency. This is crucial for further advancing the latent video diffusion models.

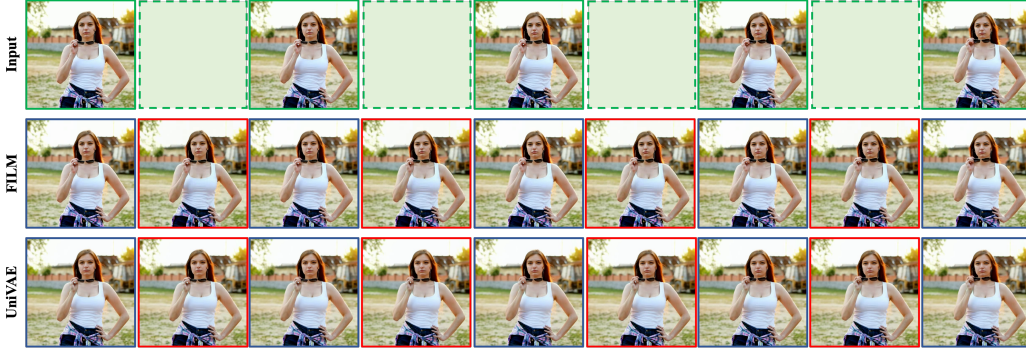


Figure 9: Performance comparison between UniVAE and FILM on frame interpolation.

Table 8: Time consumption comparison between “UniVAE” and “VAE + FILM”.

Method	UniVAE	VAE + FILM		
		VAE	FILM	Total
Time Consumption	22.83s	6.72s	464s	470.72s

## REFERENCES

- Liuhan Chen, Zongjian Li, Bin Lin, Bin Zhu, Qian Wang, Shenghai Yuan, Xing Zhou, Xinghua Cheng, and Li Yuan. Od-vae: An omni-dimensional video compressor for improving latent video diffusion model. *arXiv preprint arXiv:2409.01199*, 2024.
- Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. Unireplknet: A universal perception large-kernel convnet for audio video point cloud time-series and image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5513–5524, 2024.
- Eric Kauderer-Abrams. Quantifying translation-invariance in convolutional neural networks. *arXiv preprint arXiv:1801.01450*, 2017.
- Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
- Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision*, pp. 250–266. Springer, 2022.
- Avraham Ruderman, Neil C Rabinowitz, Ari S Morcos, and Daniel Zoran. Pooling is neither necessary nor sufficient for appropriate deformation stability in cnns. *arXiv preprint arXiv:1804.04438*, 2018.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.
- Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.
- Sijie Zhao, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Muyao Niu, Xiaoyu Li, Wenbo Hu, and Ying Shan. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*, 2024.
- Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, March 2024. URL <https://github.com/hpcaitech/Open-Sora>.