

Week 32 Report

周添文

Table of Contents

Summary of diffusion models in video generation: A Chronological Exploration

0. Generative Models for Video Generation

0.1 GANs for Video Generation

0.2 VAEs for Video Generation

1. Early T2V Exploration with Diffusion Models

1.1 Modeling Video Data

1.2 Adapting 2D UNet Architecture to 3D

1.2.1 Adapting Convolution Layers

1.2.2 Adapting Self-attention Layers

1.3 Adapting 2D DiT Architecture to 3D

1.3.1 Video Clip Patch Embedding

1.3.2 Injecting Spatio-temporal Information

1.3.3 Temporal Positional Embedding

Table of Contents

2. High-resolution Video Generation

2.1 Cascaded Generation

2.2 Latent Space Generation

3. Better frame-by-frame consistency

3.1 Noise Prior Exploration

4. Image to Video Generation

4.1 Injecting Image Condition

4.2 Enhancing Consistency with the Condition Image

5. SOTA Methods

5.1 CogVideoX

5.2 Wan-video

5.3 Sora

0. Generative Models for Video Generation

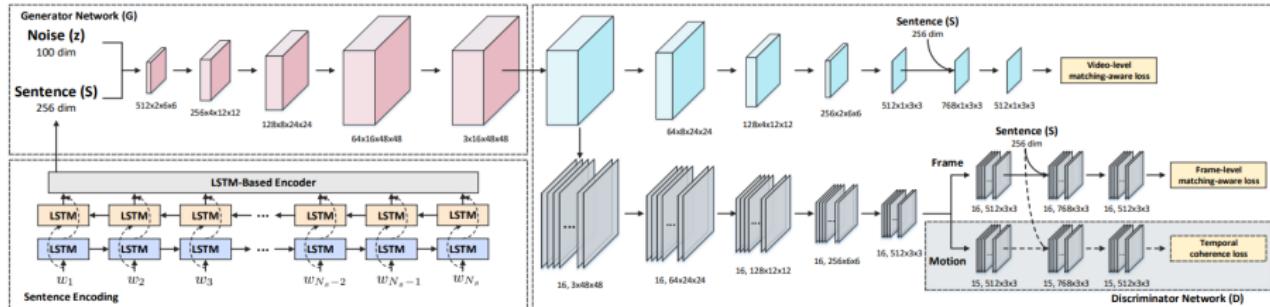
Starting Point: Generative models have been widely used in image generation tasks. As steady progress toward better image generation is made, it is also important to study the video generation problem. However, the extension from generating images to generating videos turns out to be a highly challenging task, although the generated data has just one more time dimension.

Early methods mostly train a 3D video generation model from scratch, and explicitly add controls for better generation performance.

We start by introducing early generative models for video generation in **0.1 GANs for Video Generation** and **0.2 VAEs for Video Generation**.

0. Generative Models for Video Generation

0.1 GANs for Video Generation¹

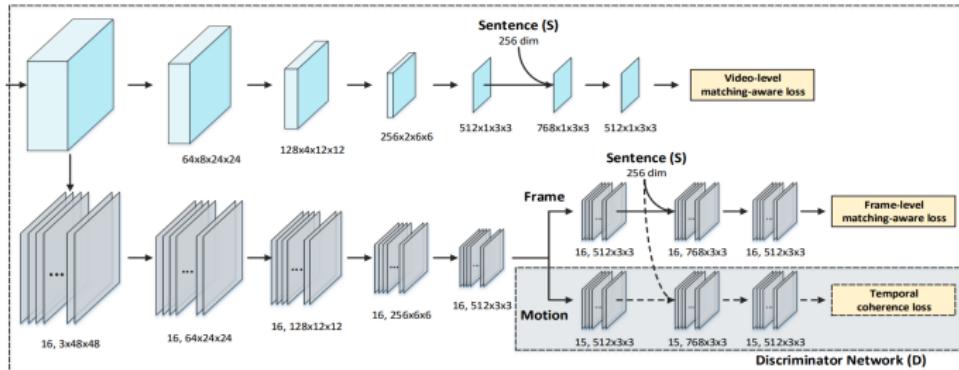


- Given a prompt, a LSTM is first utilized to embed the input word sequence, followed by a LSTM-based encoder to obtain the sentence representation S .
 - The generator network G tries to synthesize realistic videos with the concatenated input of the sentence representation S and random noise variable z .

¹Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. In Proceedings of the 25th ACM international conference on Multimedia, pp. 1789–1798. 2017.

0. Generative Models for Video Generation

0.1 GANs for Video Generation



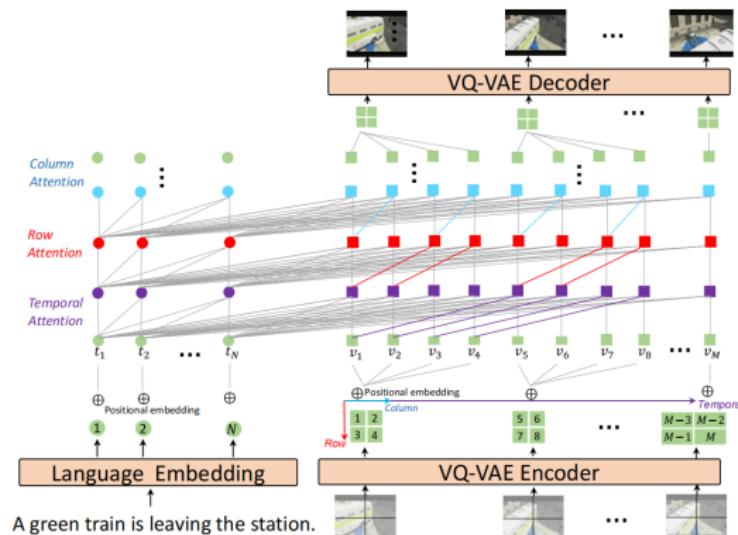
The discriminator network D includes three discriminators to **explicitly** control the generation result:

- A **video discriminator** to distinguish real video from synthetic one and align video with the correct caption
- A **frame discriminator** to determine whether each frame is real/fake and semantically matched/mismatched with the given caption.
- A **motion discriminator** to exploit temporal coherence between consecutive frames (by calculating the 2D motion tensor).

0. Generative Models for Video Generation

0.2 VAEs for Video Generation²

Adapting the 2D VQ-VAE image generation model to 3D video generation by adding a positional embedding to the discrete token encoded from each frame.



²Chenfei Wu, Lun Huang et al. Godiva: Generating open-domain videos from natural descriptions.

0. Generative Models for Video Generation

0.2 VAEs for Video Generation

Although directly adapting a 2D model to 3D can finish the generation task, the self-attention (SA) in the VQ-VAE decoder is computational expensive when dealing with longer sequences.

Hence, the author attempt to adopt a 3D sparse self-attention to reduce the computation cost when training with a auto-regressive manner.

For each latent token in position (i, j) at the $l - th$ frame, the author adapt sparsify the attention calculation as following:

$$\begin{aligned} h_{i,j,l}^{(T)} &= SA^{(T)}(v_{i,j,<l}^e), \\ h_{i,j,l}^{(R)} &= SA^{(R)}(v_{<i,j,l}^e), \\ h_{i,j,l}^{(C)} &= SA^{(C)}(v_{i,<j,l}^e). \end{aligned}$$

where T,R,C denotes temporal, row and column. $h_{i,j,l}^{(T)}, h_{i,j,l}^{(R)}, h_{i,j,l}^{(C)}$ are the hidden states at step (i, j, l) . Then, the three layers are stacked together to replace the original SA calculation:

$$h_{ijl} = \underbrace{\left[SA^{(T)}, SA^{(R)}, SA^{(C)}, SA^{(T)}, \dots, SA^{(C)} \right]}_{R \text{ layers}} (h_{<=i, <=j, <=l}),$$

1. Early T2V Exploration with Diffusion Models

Starting point: Adapting image generation models to video generation models.

Main Adaption: Accept 3D input (video sequence), produce 3D output (with multiple coherent frames), instead of single 2D images.

2D output → 3D output

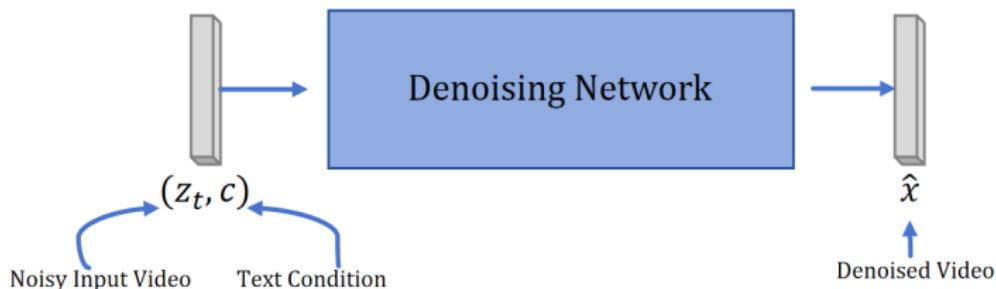


To achieve such adaption, there should be changes in **1.1 Modeling Video Data**, and **1.2 Adapting 2D Architecture to 3D**.

1. Early T2V Exploration with Diffusion Models

1.1 Modeling Video Data:

Drawing inspiration from 2D image diffusion models, early attempts (e.g. VDM³) adopt the following method for modeling video data with text prompt.



³ Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. In arXiv:2204.03458, 2022b.

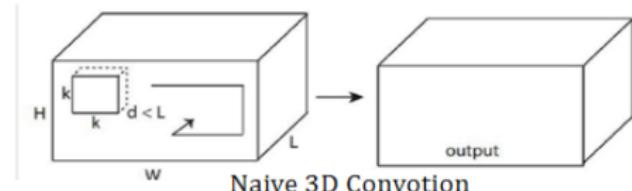
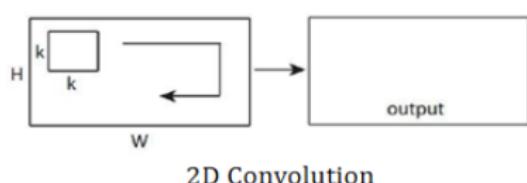
1. Early T2V Exploration with Diffusion Models

1.2. Adapting 2D UNet Architecture to 3D UNet:

With the change in modeling, the architecture we adopt, i.e. the 2D Denoising UNet, should also be changed to process 3D data, especially the Convolution layers and self-attention layers.

1.2.1 Adapting Convolution Layers:

The most straightforward way is to adopt a naive 3D ⁴ convolution kernel that convolves across the spatial dimensions as well as the temporal dimension.



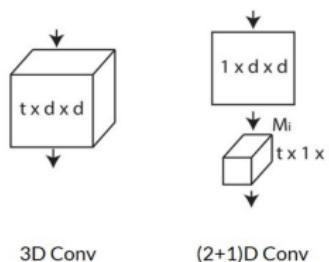
⁴Du et al., “Learning Spatiotemporal Features with 3D Convolutional Networks,” ICCV 2015.

1. Early T2V Exploration with Diffusion Models

1.2.1 Adapting Convolution Layers:

The naive 3D convolution layer is:

1. Computational demanding with increased the number of parameters computational complexity.
2. Treats spatial and temporal dimensions equally, which may not be optimal.

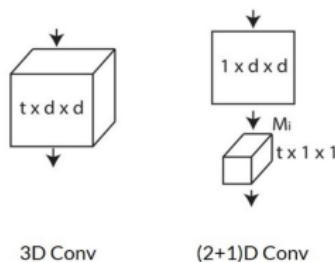


Hence, (2+1)D convolution⁵, namely pseudo 3D convolution, is proposed.

⁵Du et al., “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” CVPR 2018.

1. Early T2V Exploration with Diffusion Models

1.2.1 Adapting Convolution Layers:



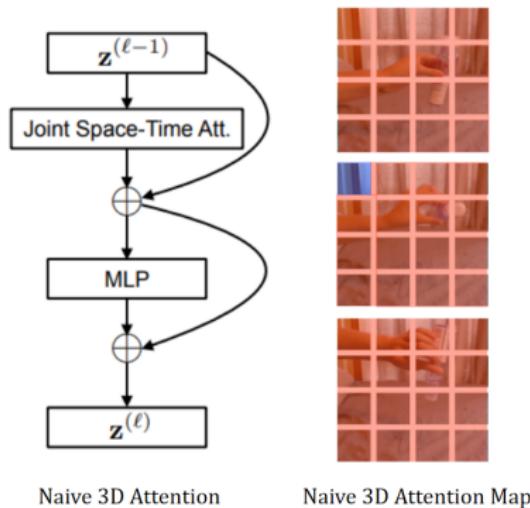
The (2+1)D Convolution decomposes the 3D convolution into a 2D spatial conv and a 1D temporal convolution, hence bringing the following benefits:

1. Reduces the computational cost and number of parameters.
2. Doubles the non-linearity in the model (1 ReLU after each kernel).
3. Enables using pretrained weights from image diffusion models for the 2D spatial conv.

1. Early T2V Exploration with Diffusion Models

1.2.2 Adapting Self-attention Layers:

The easiest way to adapt self-attention layers is also by simply adding a temporal dimension to the original 2D self-attention, namely naive 3D self-attention⁶, as shown below:



All of the frames (red patches) are included in calculation.

⁶Gedas et al. Is space-time attention all you need for video understanding. arXiv

1. Early T2V Exploration with Diffusion Models

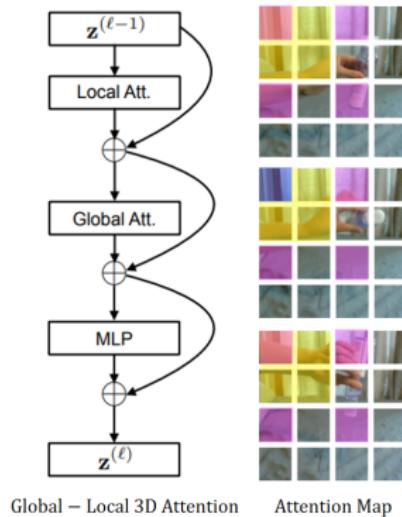
1.2.2 Adapting Self-attention Layers:

However, such naive approach unsurprisingly yield a heavy computational burden. Hence, several **decomposing** strategies, and **sparsifying** strategies (common approach in enhancing attention efficiency) are proposed.

1. Early T2V Exploration

1.2.2 Adapting Self-attention Layers:

Strategy 1: Global-Local Attention

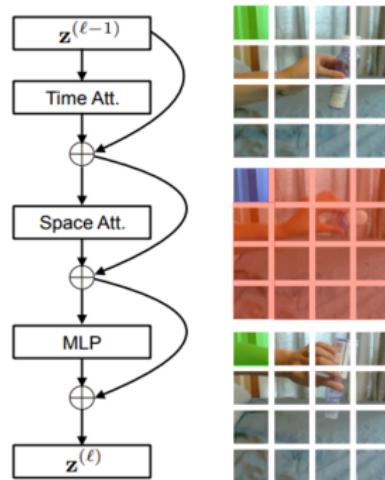


Can be viewed as a faster approximation of the naive 3D attention using a local-global decomposition and a sparsity pattern.

1. Early T2V Exploration with Diffusion Models

1.2.2 Adapting Self-attention Layers:

Strategy 2: Divided Time-Space Attention



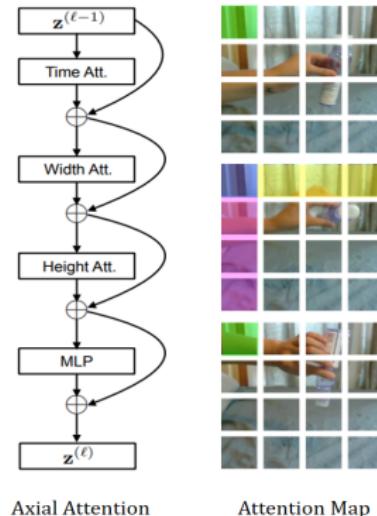
Divided Space – Time Attention Attention Map

Decompose the naive 3D attention to a spatial and temporal dimension.

1. Early T2V Exploration

1.2.2 Adapting Self-attention Layers:

Strategy 3: Axial Attention



Decompose the naive 3D attention over time, width and height

1. Early T2V Exploration with Diffusion Models

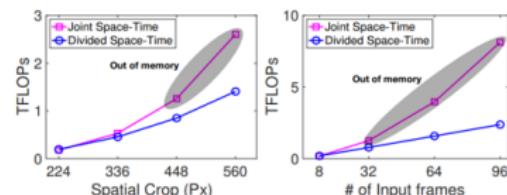
1.2.2 Adapting Self-attention Layers:

The decomposition provides computation efficiency and better learning capacity (more parameters), while too much decomposing might damage the captured temporal and spatial information.

The divided time-space attention achieves the best balance in this trade-off.

Attention	Params	K400	SSv2
Space	85.9M	76.9	36.6
Joint Space-Time	85.9M	77.4	58.5
Divided Space-Time	121.4M	78.0	59.5
Sparse Local Global	121.4M	75.9	56.3
Axial	156.8M	73.5	56.2

Divided Space-Time Attention achieves the best video-level accuracy with larger learning capacity



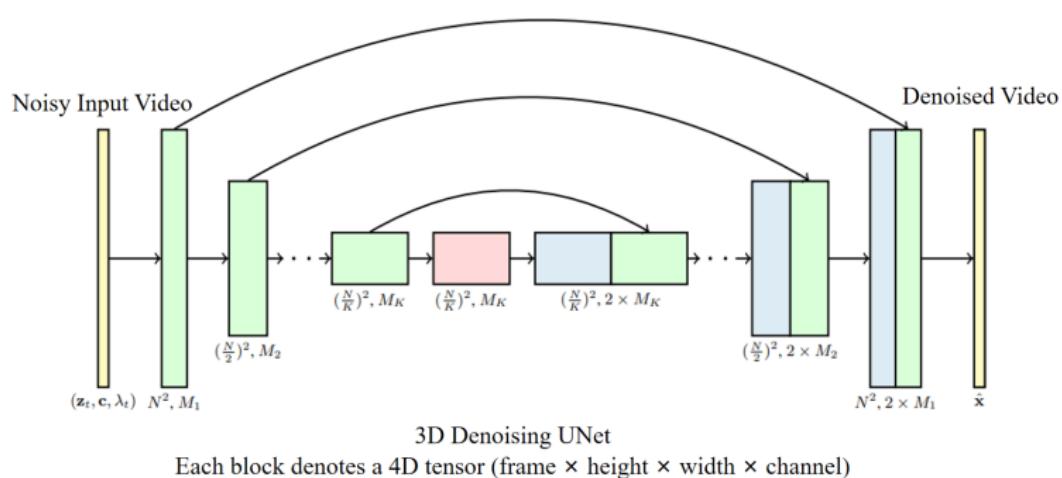
Naive 3D attention (Joint Space-time) requires large computation cost when processing high resolution long videos

1. Early T2V Exploration with Diffusion Models

1.2. Adapting 2D UNet Architecture to 3D:

Hence, VDM proposed a 3D denoising UNet derived from 2D denoising UNet as shown below.

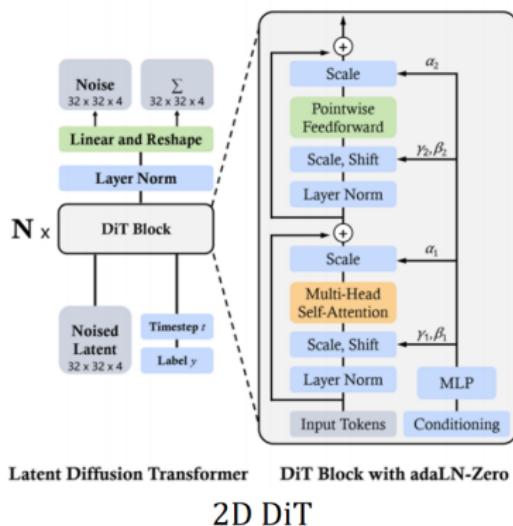
The 2D convolution layers are adapted to the pseudo 3D convolution, and a temporal attention layer is inserted after each spatial attention layer to achieve divide space-time attention.



1. Early T2V Exploration with Diffusion Models

1.3. Adapting 2D DiT Architecture to 3D:

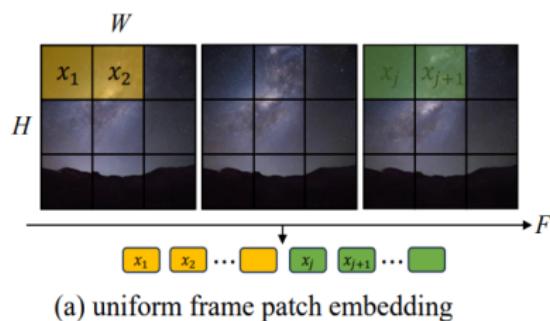
Another line of work utilize Diffusion Transformer (DiT) framework for image generation. To adapt the 2D DiT to capture the spatio-temporal information, we introduce the adaptions to the 2D DiT architecture from patch embedding, block architecture, and temporal positional embedding.



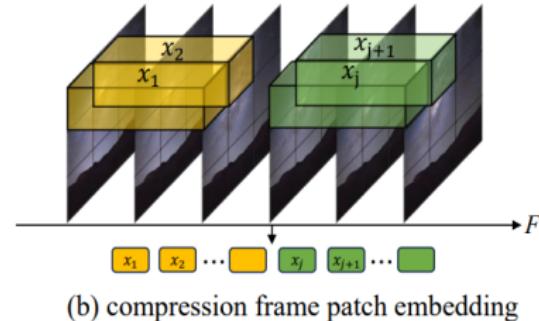
1. Early T2V Exploration with Diffusion Models

1.3.1. Video Clip Patch Embedding

Different from the UNet architecture, Transformers can naturally handle 3D input since they accept them as tokens. Now, we consider the patch embedding mechanism that embed the input frames into tokens. Latte⁷ attempted 2 ways for the embedding, as shown below.



(a) uniform frame patch embedding



(b) compression frame patch embedding

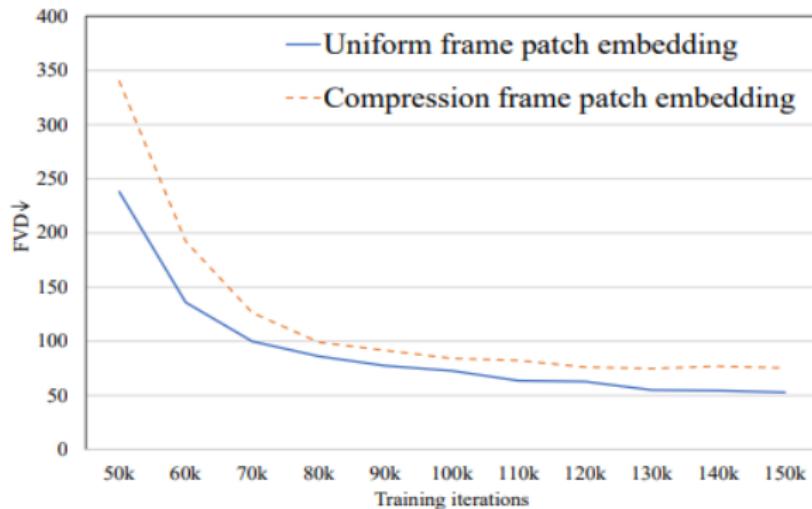
The embedding is employed in the latent space, only the visualization is in pixel space.

⁷Latte: Latent Diffusion Transformer for Video Generation

1. Early T2V Exploration with Diffusion Models

1.3.1. Video Clip Patch Embedding

After conducting the ablation study on FVD (Video version FID), the result favors the uniform frame patch embedding strategy, since the temporal dimension compression might yield temporal information loss in the input frames.



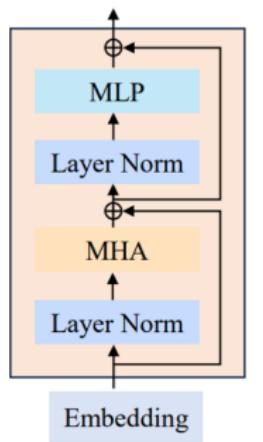
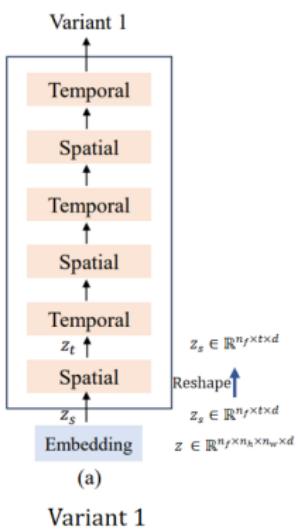
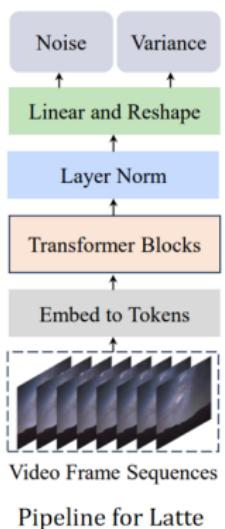
1. Early T2V Exploration with Diffusion Models

1.3.2. Injecting Spatio-temporal Information

The author surveyed through several methods to capture the spatio-temporal information. The primary idea is also to **decompose** the spatial and temporal information separately either in a block-wise manner or an inner-block manner.

1. Early T2V Exploration with Diffusion Models

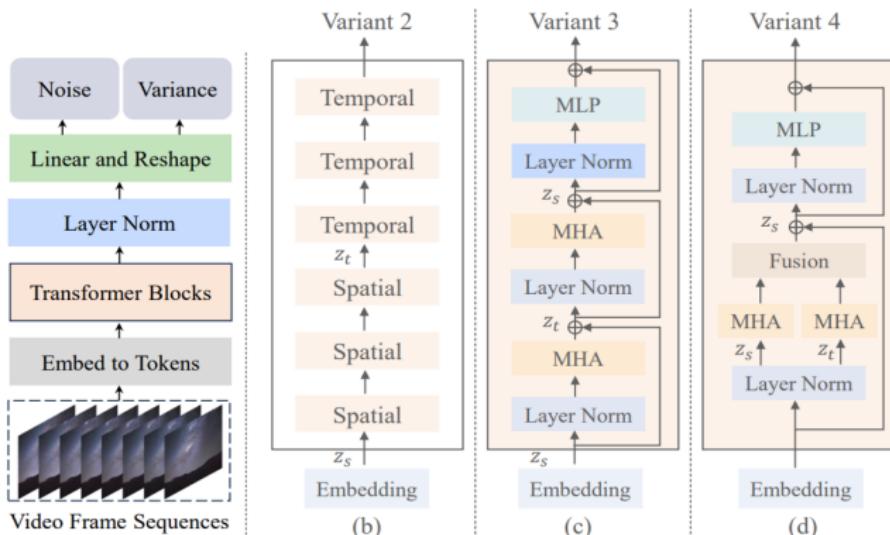
1.3.2. Injecting Spatio-temporal Information



Variant 1 decompose the spatial and temporal Transformer in an interleaved manner. The feature vector is reshaped between each spatial and temporal transformer.

1. Early T2V Exploration with Diffusion Models

1.3.2. Injecting Spatio-temporal Information

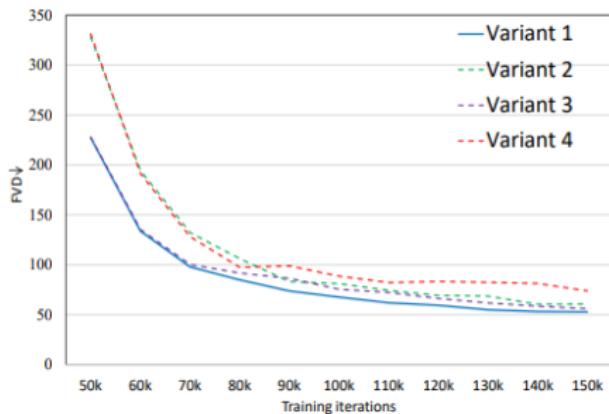


Apart from Variant 1, the author also proposes 3 variant with different decomposition strategies (late fusion, inner-block decomposition).

1. Early T2V Exploration with Diffusion Models

1.3.2. Injecting Spatio-temporal Information

The author also conduct ablation study on the 4 proposed variant, and Variant 1 yields the best result, showing the superiority of the interleaved method.



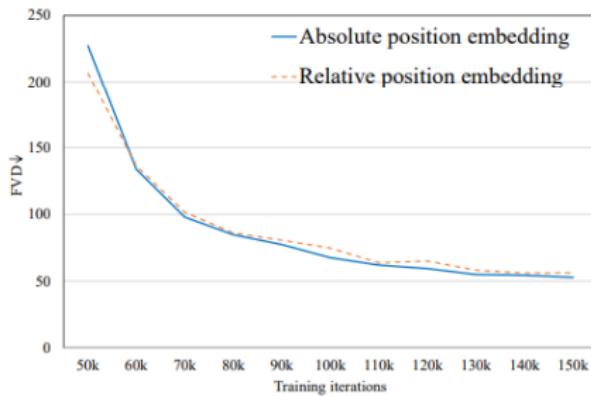
1. Early T2V Exploration with Diffusion Models

1.3.3. Temporal Positional Embedding

To help the model learn the temporal relationship between frames, an additional temporal positional embedding is added to the input of the transformer blocks.

Two ways of embedding are compared, absolute embedding (sin and cos of different frequencies) and RoPE (rotary positional embedding).

Ablation study shows that absolute embedding achieves better result.



1. Early T2V Exploration with Diffusion Models

1.3.3. Temporal Positional Embedding

In a nutshell, Latte adopts the best setting of each module in their ablation study, and formed a DiT based video generation pipeline, which is similar to the one adopted by Sora⁸.

⁸OpenAI, “Sora: Creating video from text.” <https://openai.com/sora>, 2024.

2. High-resolution Video Generation

Starting Point: The above method successfully achieves text to video generation, but it relies heavily on paired text-video data, which is relatively scarce, since the model is trained from scratch. Hence, image pretraining techniques are proposed to leverage the text-to-image models pretrained on large scale paired image dataset. The two main approaches to utilize pretrained T2I models for video generation tasks are **2.1 Cascaded Generation**, and **2.2 Latent Space Generation**.

2. High-resolution Video Generation

2.1 Cascaded Generation

Structure: Cascaded generation starts from generating key frames from a T2V model, and apply Temporal Super Resolution (TSR) and Spatial Super Resolution (SSR) in an **interleaved manner** with in a **non-overlapping window**. These models **can be initialized with pretrained T2I models**, and **only require minimal fintuning on video dataset**.

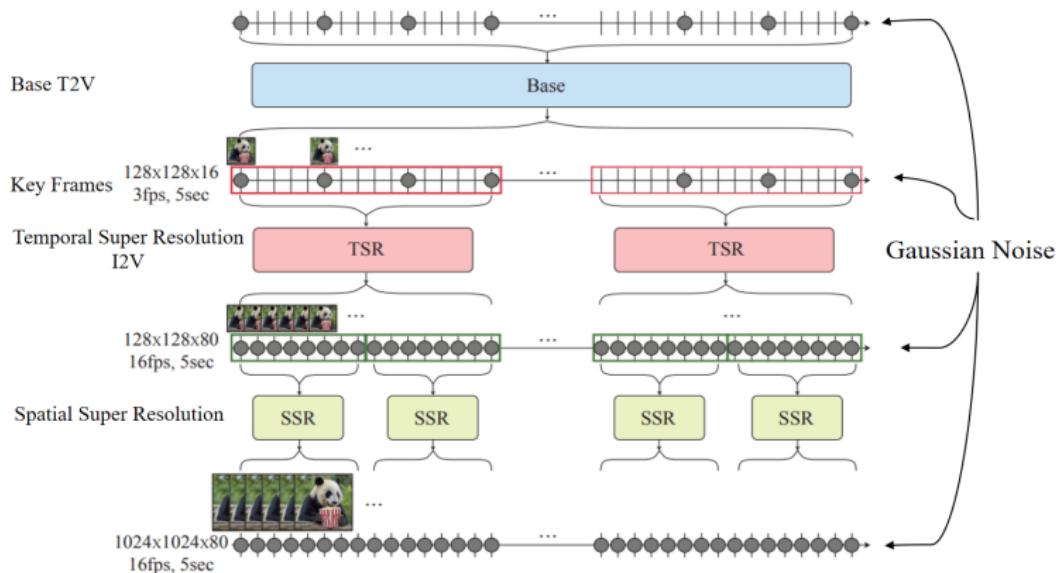
Interleaved manner: The interleaved window-based operation provides a trade-off between the computational efficiency and the temporal coherence within the generated frames.

A commonly adopted pipeline is shown below:

2. High-resolution Video Generation

2.1 Cascaded Generation

Common pipeline of cascaded generation:

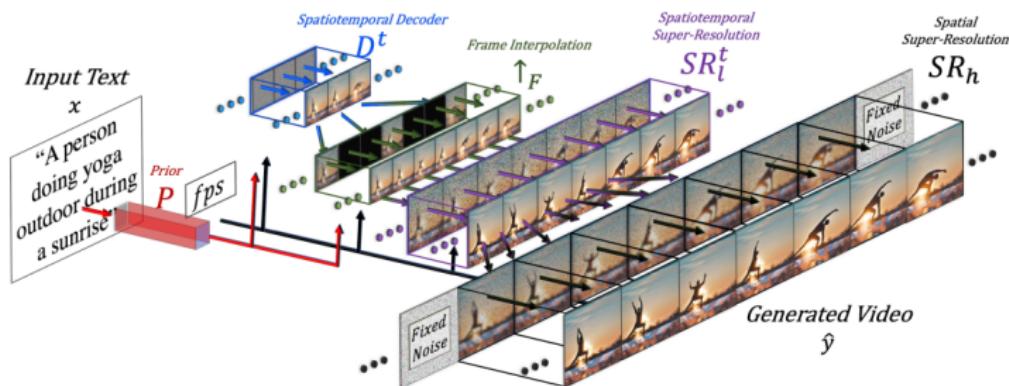


2. High-resolution Video Generation

2.1 Cascaded Generation

The main benefit of this cascade generation manner is the ability to generate long and high resolution video, with pretrained T2I models as prior knowledge.

Make-a-Video⁹, as a milestone in cascaded generation, achieves the goal with the strategies below.



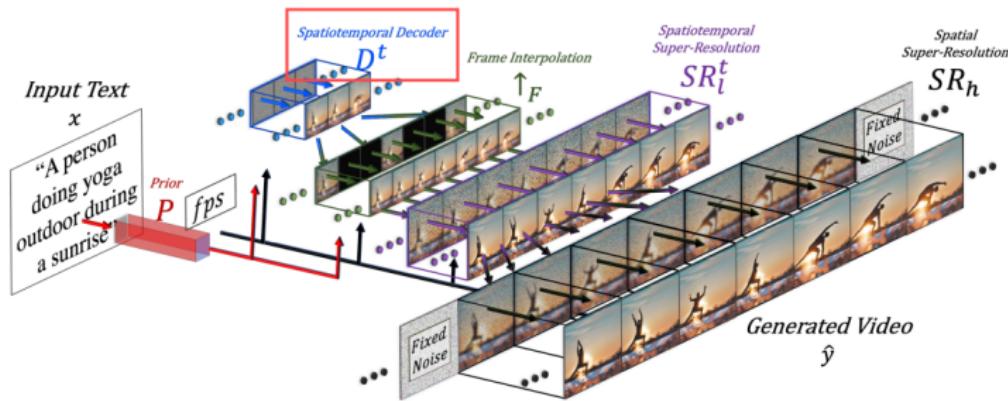
⁹Singer et al., “Make-A-Video: Text-to-Video Generation without Text-Video Data,” arXiv 2022.

2. High-resolution Video Generation

2.1 Cascaded Generation

This framework utilize image pretraining by initializing the **decoder** D_t with a frozen pretrained T2I model, which accept CLIP Embedding as input.

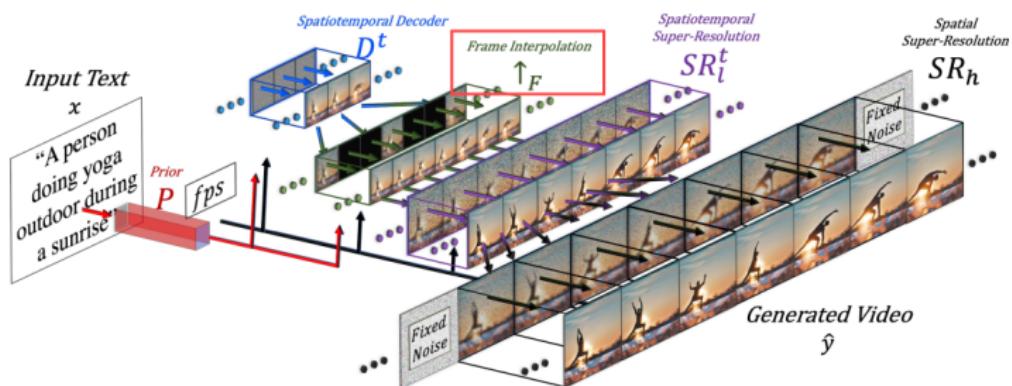
Then, temporal layers (i.e. divided space-time attention, (2+1)D Convolution) are inserted, and **only the temporal layers** are finetuned on minimal video dataset, yielding a spatiotemporal decoder D_t .



2. High-resolution Video Generation

2.1 Cascaded Generation

The frame interpolation model F is further finetuned on the spatiotemporal decoder D_t , with the target adjusted to frame interpolation.

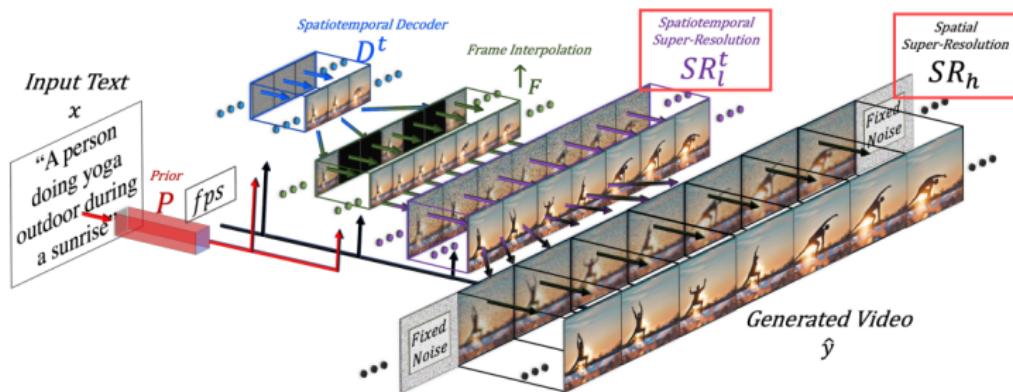


While finetuning, the masked video and a binary mask indicating which frame is masked, is concatenated to the UNet's input, serving as condition.

2. High-resolution Video Generation

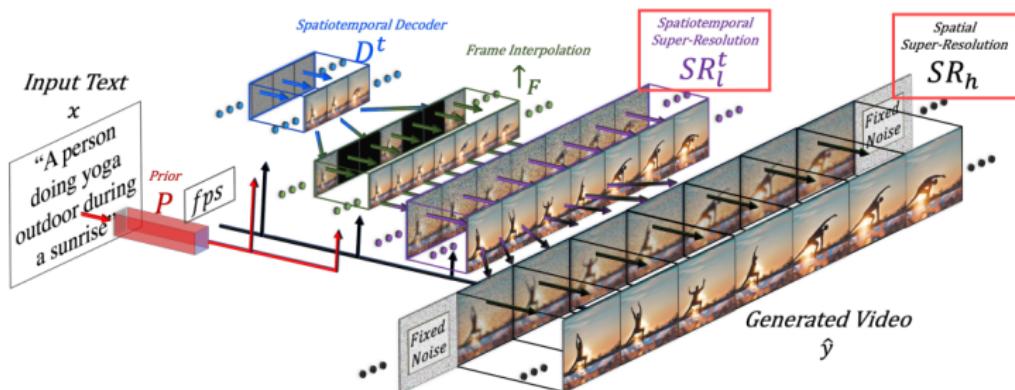
2.1 Cascaded Generation

Ideally, we should implement the spatial super-resolution with **temporal information** as the previous strategies, since super-resolution will result in hallucination, hence yielding a flickering effect within frames.



2. High-resolution Video Generation

2.1 Cascaded Generation

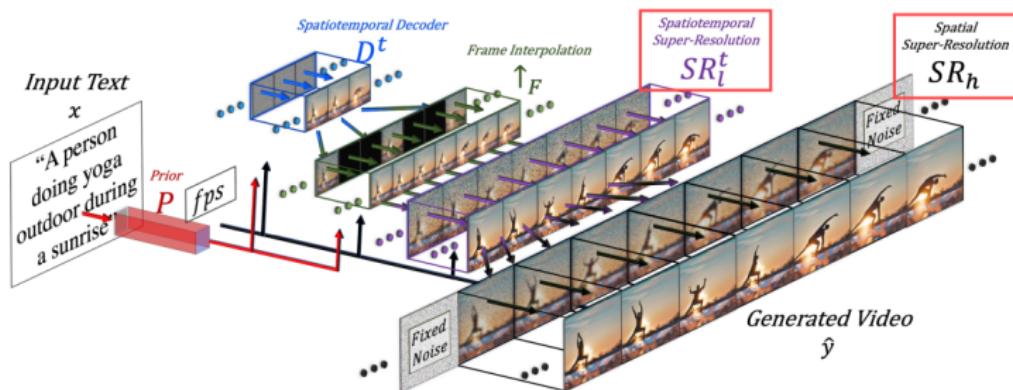


However, due to the computational constraints, involving temporal attention in high resolution space is computational demanding.

To strike such balance, the author implement two super-resolution models, one SR_l^t with temporal information involved, in a lower resolution space; and one SR_h without temporal information, in high resolution space.

2. High-resolution Video Generation

2.1 Cascaded Generation

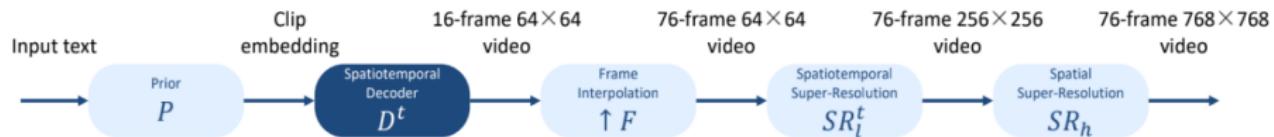


In this stage, the author also successfully leverage image pretraining by pretraining the two SR models on image datasets first, with the task of super-resolution from downsampled images.

Then, temporal layers are inserted into SR_l^t , and are again finetuned on minimal video data.

2. High-resolution Video Generation

2.1 Cascaded Generation



In a nutshell, all four networks have successfully involved prior knowledge via image pretraining, using the above strategies.

2. High-resolution Video Generation

2.2 Latent Space Generation

Another way to leverage image pretraining is via latent space generation with finetuning on the pre-trained Latent Diffusion Models (LDM).

A naive way of adapting pretrained LDM for video generation is as following, which merely involves inserting temporal layers in the pretrained latent denoising UNet, while finetuning on video dataset. The VAE encoder and decoder are directly from the image LDM.

2. High-resolution Video Generation

2.2 Latent Space Generation

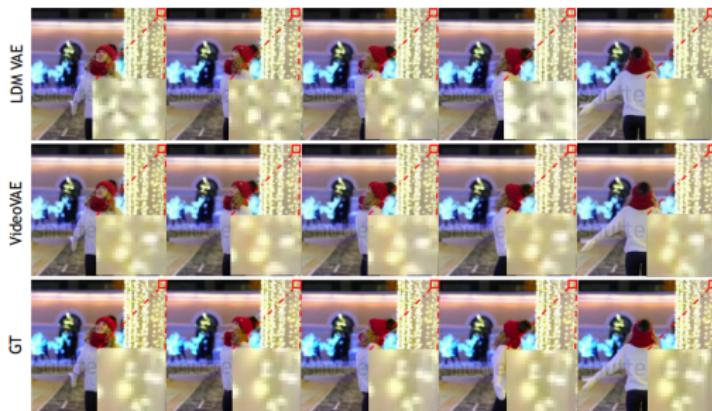
Such naive adaption successfully leverage the pretrained LDM, which achieves our goal of image pretraining, but it still suffers from the problem below:

1. The VAE decoder is pretrained with text-image pair, hence, directly adopting the pretrained VAE Decoder will result in flickering artifacts.
2. The VAE compression is only conducted on spatial dimension, but the redundancy in temporal dimension is not compressed.

2. High-resolution Video Generation

2.2 Latent Space Generation

MagicVideo¹⁰ visualize the flickering artifacts between generated frames of the naive method, as shown in row 1 and 3 below.

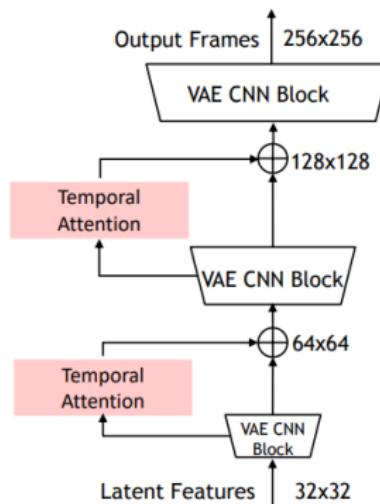


¹⁰Zhou, Daquan, et al. "Magicvideo: Efficient video generation with latent diffusion models." arXiv preprint arXiv:2211.11018 (2022).

2. High-resolution Video Generation

2.2 Latent Space Generation

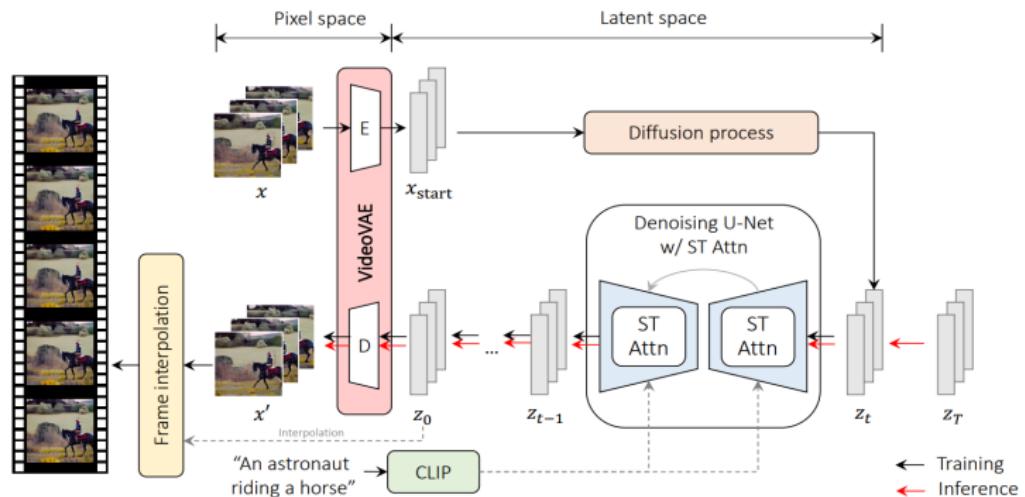
To mitigate such problem, the author propose a Video VAE decoder, to substitute the original VAE Decoder. The temporal attention adopted here is also the divide space-time attention, which we have previously mentioned in Section 1.2.2.



2. High-resolution Video Generation

2.2 Latent Space Generation

With the proposed Video VAE decoder, the framework is shown as below:

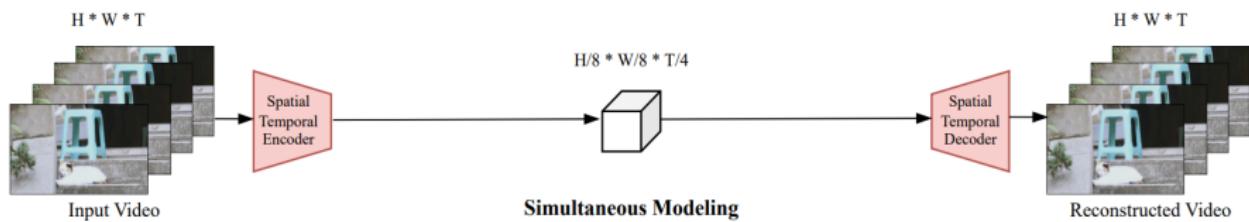


2. High-resolution Video Generation

2.2 Latent Space Generation

To deal with the second problem, i.e. to achieve compression on both temporal dimension and spatial dimension, VideoVAE¹¹ propose a 3D VAE autoencoder.

Two kinds of settings are explored by the author:



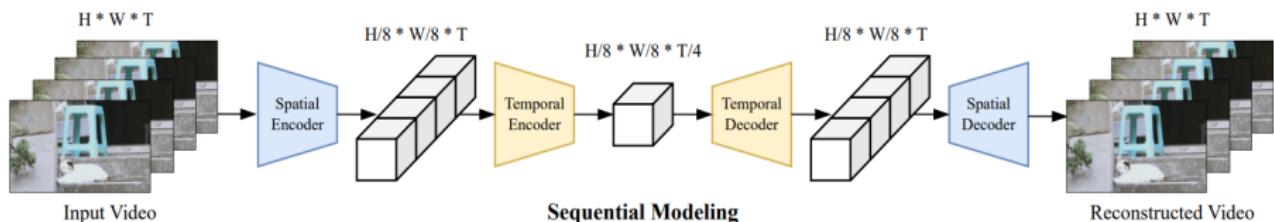
The simultaneous modeling is achieved by replacing the 2D convolution in image VAE with pseudo (2+1)D convolution, and turning the spatial attention layer into a spatiotemporal attention layer by inserting temporal attention. with the same strategy in 1.2.2.

¹¹Xing, Yazhou, et al. "Large Motion Video Autoencoding with Cross-modal Video VAE." arXiv preprint arXiv:2412.17805 (2024).

2. High-resolution Video Generation

2.2 Latent Space Generation

The sequential modeling process includes first utilizing the image VAE encoder to compress the input video frame-by-frame . Then learn a light-weighted temporal autoencoding process to further compress the temporal redundancy.

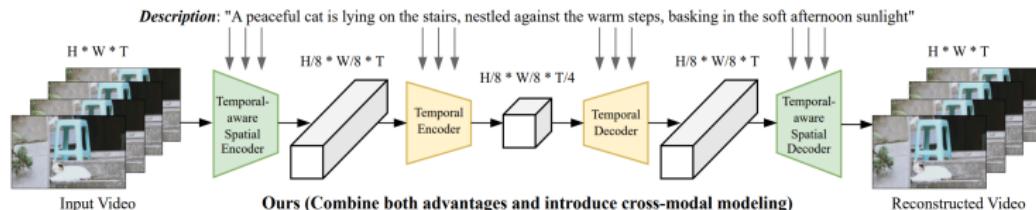


The authors find that the sequential spatio-temporal design can better compress and recover the dynamic of the input video than the simultaneous design, but is not good at recovering spatial details.

2. High-resolution Video Generation

2.2 Latent Space Generation

Hence, the solution is to combine the benefits of both approaches.



The input video is first compressed spatially by the Temporal-aware Spatial Encoder, which is initialized from image VAE, and finetuned with video data. The adaption is similar to Section 1.2.2.

Then the spatially compressed vector is compressed temporally with a light weight temporal autoencoder.

2. High-resolution Video Generation

2.2 Latent Space Generation

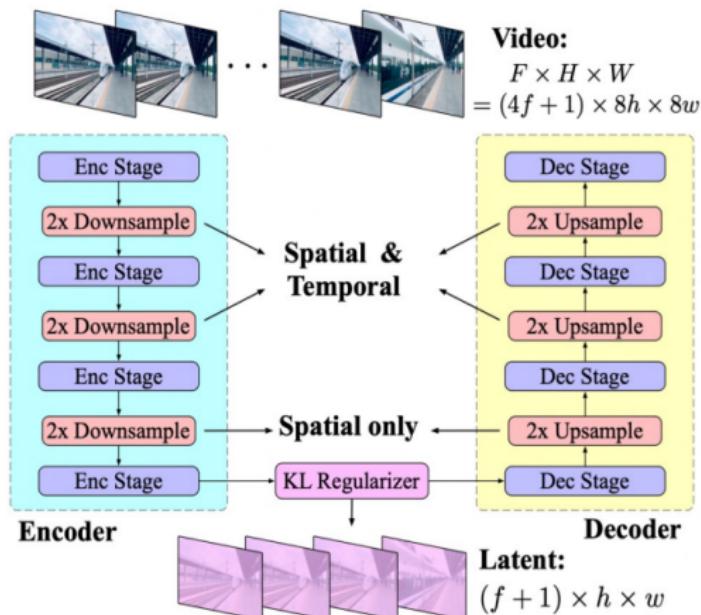
Such a design lead to a better compression and reconstruction result for video VAE, as shown in row 1 and 4 below:



2. High-resolution Video Generation

2.2 Latent Space Generation

Another commonly adopted 3D VAE autoencoder is proposed by CogVideoX¹² to achieve x4 downsample temporally and x8 spatially.



¹²CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer ▶

2. High-resolution Video Generation

2.2 Latent Space Generation

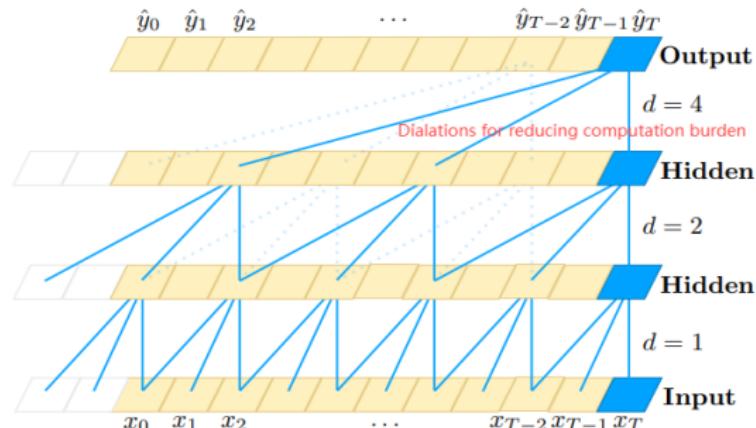
This framework train a **3D temporal causal convolution** from scratch to substitute the original 2D convolution in the down/upsample blocks.

Compared to the naive 3D convolution we have proposed in Section 1.2.1, the temporal dimension is convolved in a causal manner.

2. High-resolution Video Generation

2.2 Latent Space Generation

The motivation of this causal manner comes from the problem encountered in temporal convolution, where the future information should not influence the present or past predictions¹³, similar to RNNs.



where the convolution result of y_t only consider its previous frames y_0 to y_{t-1} .

¹³An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling

2. High-resolution Video Generation

2.2 Latent Space Generation

Assume the kernel size on temporal dimension is k_t , such causal 3D convolution layer pads with $k_t - 1$ frames **before the input** and nothing after, instead of $\lfloor \frac{k_t-1}{2} \rfloor$ before and $\lfloor \frac{k_t}{2} \rfloor$ after. so that the output for each frame **only depends on the previous frames**.

With such padding strategy, the first frame is always independent of other frames, allowing the model to tokenize single images.

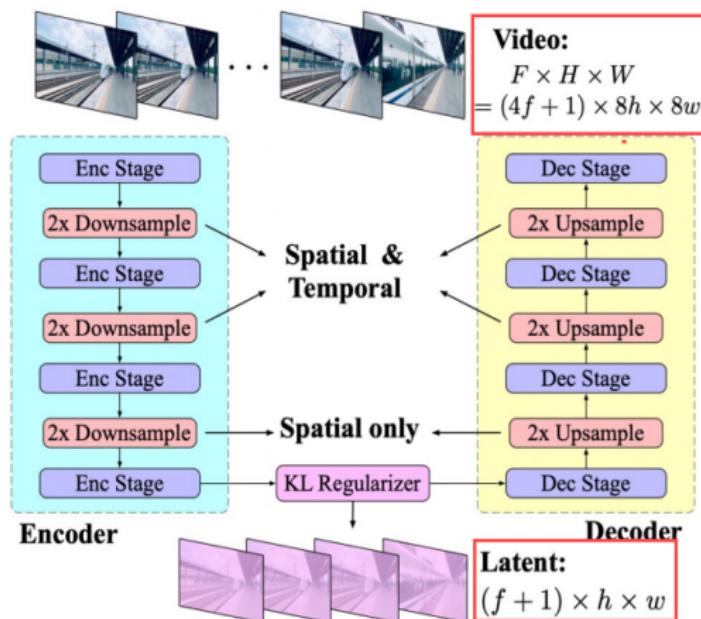
Hence enabling the model to realize image-video co-training, which is proved¹⁴ to be an effective strategy in training T2V models.

¹⁴ Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J.Fleet. Video Diffusion Models. In arXiv:2204.03458, 2022b.

2. High-resolution Video Generation

2.2 Latent Space Generation

According to this padding strategy, assuming the compressed latent has size $(f + 1) \times h \times w$, the input to this 3D temporal causal VAE Encoder should be $(4f + 1) \times 8h \times 8w$



3. Better Frame-by-frame Consistency

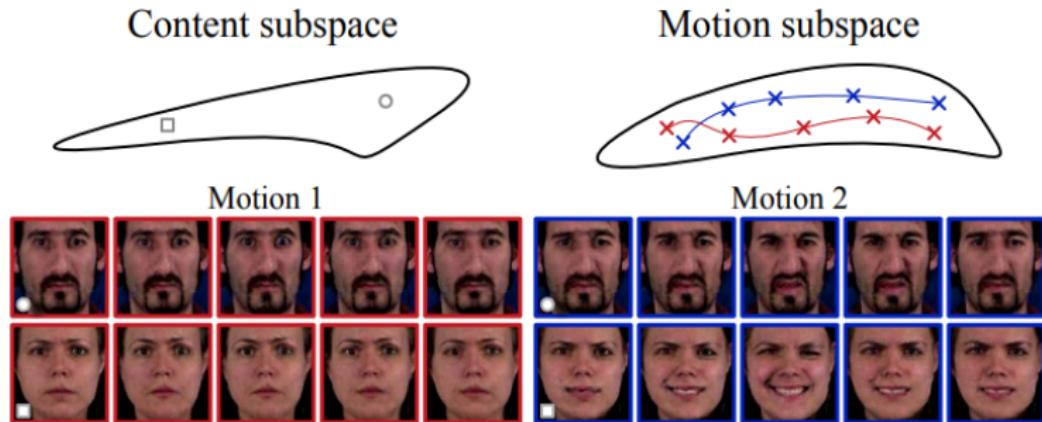
Starting Point: To achieve better frame-by-frame consistency in the generated image, (nearby) frames in a video clip should be almost the same in content (background, person's identity etc.), the main difference should be the motion within frames.

To enhance frame-by-frame consistency, we introduce **3.1 Noise Prior Exploration**.

3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

Since visual signals in a video can be divided into content and motion, MoCoGAN¹⁵ is a pioneer work to propose that the latent space of GAN for video generation can be decomposed to a **content subspace** and a **motion subspace**.

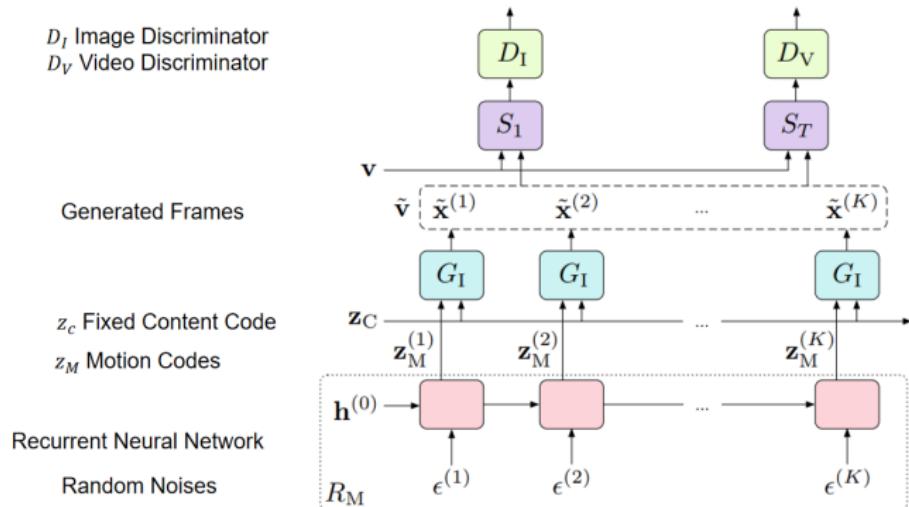


¹⁵MoCoGAN: Decomposing Motion and Content for Video Generation

3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

Hence, during the sampling process, the once the content latent code is fixed, the generated video exhibits great temporal coherence among the frames.



3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

Borrowing such an idea, Video Fusion¹⁶ decomposes the Gaussian Noise in the sampling process of Video Diffusion Model into a base noise and a residual noise.

The base noise contains the content of the image, so adopting **the same base noise** across all the frames ensures consistency in the content among the frames.

The noise removal task can be solved by subsequently removing the base noise and the residual noise. Since the base noise is removed, the residual noise will be lighter, and will be easier to get removed.

¹⁶VideoFusion: Decomposed Diffusion Models for High-Quality Video Generation



3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

- Base Noise Generator:

In each diffusion step t , the noised latent z_i^t for the $i - th$ frame can be written as:

$$z_i^t = \sqrt{\hat{\alpha}_t}x_i + \sqrt{1 - \hat{\alpha}_t}\epsilon_i^t \quad (1)$$

And each frame can be decomposed into a base frame x_0 with a residual Δx_i :

$$x_i = \sqrt{\lambda_i}x_0 + \sqrt{1 - \lambda_i}\Delta x_i, \quad i = 1, 2, \dots, N \quad (2)$$

yielding

$$z_i^t = \sqrt{\hat{\alpha}^t} \left(\sqrt{\lambda_i}x_0 + \sqrt{1 - \lambda_i}\Delta x_i \right) + \sqrt{1 - \hat{\alpha}^t}\epsilon_i^t \quad (3)$$

3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

- Base Noise Generator:

As stated before, the noise can also be decomposed into a base noise and a residual noise, as below:

$$\epsilon_i^t = \sqrt{\lambda_i} b_i^t + \sqrt{1 - \lambda_i} r_i^t \quad b_i^t, r_i^t \sim \mathcal{N}(0, 1) \quad (4)$$

Substitute Eq.(4) into Eq.(3), we yield:

$$z_i^t = \underbrace{\sqrt{\lambda_i} \left(\sqrt{\hat{\alpha}^t} x_0 + \sqrt{1 - \hat{\alpha}^t} b_i^t \right)}_{\text{diffusion of } x_0} + \underbrace{\sqrt{1 - \lambda_i} \left(\sqrt{\hat{\alpha}^t} \Delta x_i + \sqrt{1 - \hat{\alpha}^t} r_i^t \right)}_{\text{diffusion of } \Delta x_i} \quad (5)$$

In this way, x_0 in different frames will be noised to the same value.

3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

- Base Noise Generator:

Since the base noise b_t is shared among all the frames in a same timestep, we can further write Eq.(5) as:

$$z_i^t = \sqrt{\hat{\alpha}^t} x_i + \sqrt{1 - \hat{\alpha}^t} \left(\sqrt{\lambda_i} b_t + \sqrt{1 - \lambda_i} r_i^t \right) \quad (6)$$

Now, if we select a frame as the base frame, e.g. frame $i = \lfloor N/2 \rfloor$, we yield:

$$z_i^t = \begin{cases} \sqrt{\hat{\alpha}_t} x^i + \sqrt{1 - \hat{\alpha}_t} b_t & \text{if } i = \lfloor N/2 \rfloor \\ \sqrt{\hat{\alpha}_t} x^i + \sqrt{1 - \hat{\alpha}_t} (\sqrt{\lambda_i} b_t + \sqrt{1 - \lambda_i} r_i^t) & \text{if } i \neq \lfloor N/2 \rfloor \end{cases} \quad (7)$$

Hence, as long as we feed frame $x^{\lfloor N/2 \rfloor}$ into any off-the-shelf noise predicting denoiser z_ϕ^b , we get the base noise b_t .

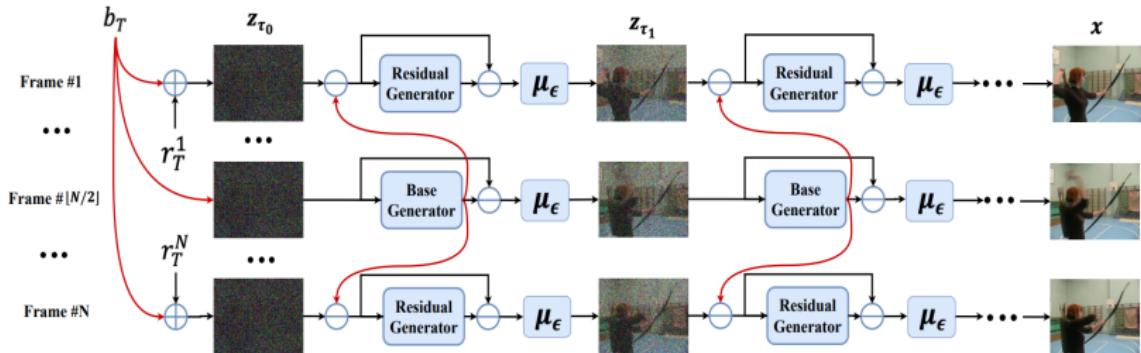
3. Better Frame-by-frame Consistency

3.1 Noise Prior Exploration

- Residual Noise Generator:

After subtracting the base noise from each frame, the residual noise is much more lighter. Hence, the author train a separate residual noise estimation network to generate the residual noise r_i^t for each frame in each timestep.

Then, the result can be generated with a DDIM sampler.



4. Image to Video Generation

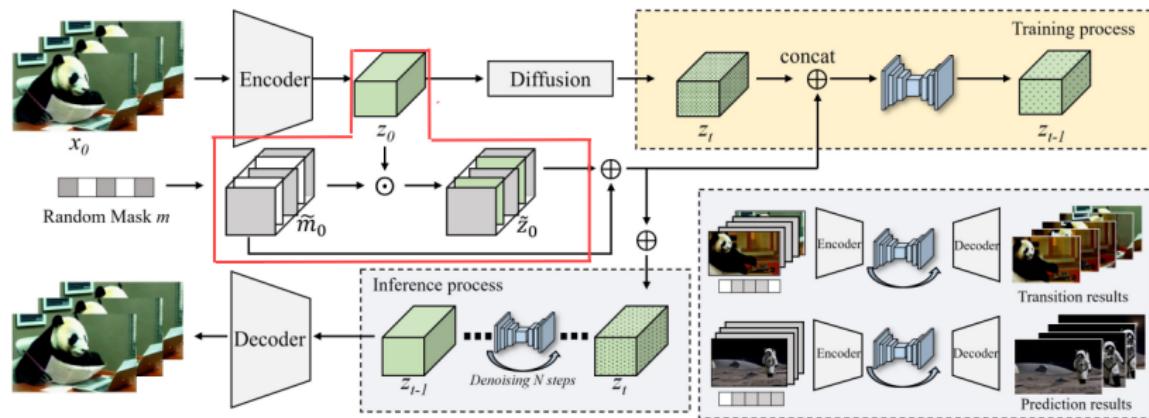
Except for text, images can also be served as a source of condition for video generation. The I2V input could be a **single image**, which is commonly chosen as first frame, or a **short video**, which can be seen as a frame interpolation framework.

Since the image is an additional condition for the model, the modeling should be altered correspondingly. We now introduce **4.1 Injecting image condition** and **4.2 Enhancing consistency with the condition image**.

4. Image to Video Generation

4.1 Injecting Image Condition

A common approach¹⁷ to modeling the I2V problem is shown as below:



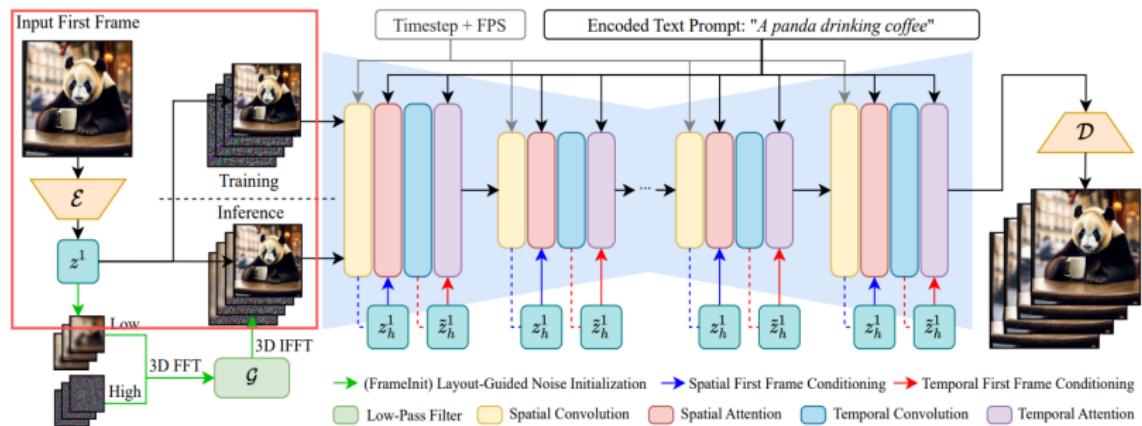
During training, the randomly masked latent z_0 is **concatenated** to each noisy z_t as condition. While during inference, the masked (latent) input is also concatenated to the denoised z_t as condition.

¹⁷SEINE: Short-to-Long Video Diffusion Model for Generative Transition and Prediction

4. Image to Video Generation

4.1 Injecting Image Condition

Another approach¹⁸ to injecting image condition is by directly substituting the initial noise of first frame by the given input frame both in training and inference, as shown below:



¹⁸ConsistI2V: Enhancing Visual Consistency for Image-to-Video Generation

4. Image to Video Generation

4.2 Enhancing consistency with the condition image

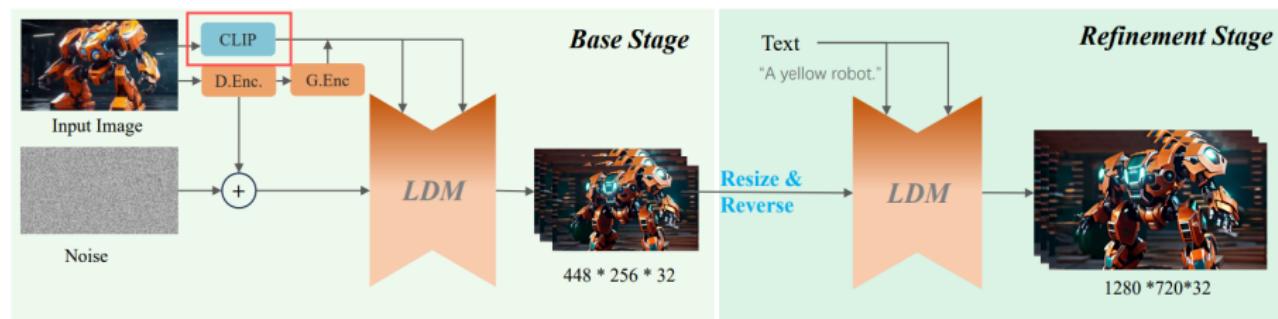
Although the above pipelines successfully achieve the I2V task, better generation result can be achieved by improving semantic and low-level consistency with the input frame.

The core idea here is to **fully leverage** the information in the input image, and **effectively inject** them into the generation process.

4. Image to Video Generation

4.2 Enhancing consistency with the condition image

I2VGen-XL¹⁹ propose to first inject the high-level semantic features extracted by a CLIP image encoder via a cross-attention into the VideoLDM (as stated in Section 2.2).



However, such approach resulted in poor preservation of the content and structure of the input image in the generated videos because the semantic control is relatively weak, since CLIP is pretrained on aligning visual and language features, disregards the perception of fine details in the images.

¹⁹I2VGen-XL: High-Quality Image-to-Video Synthesis via Cascaded Diffusion Models ↗ ↘

4. Image to Video Generation

4.2 Enhancing consistency with the condition image



Hence, the author further propose a detail encoder (D.Enc) and a global encoder (G.Enc).

The D.Enc is selected to be a VQGAN Encoder that effectively extract the **low-level features** of the input image, and the extracted features are added to the initial noise of each frame.

The G.Enc is a multi-scale feature extractor, that assist the CLIP Encoder in injecting **semantic features** to the LDM via cross-attention.

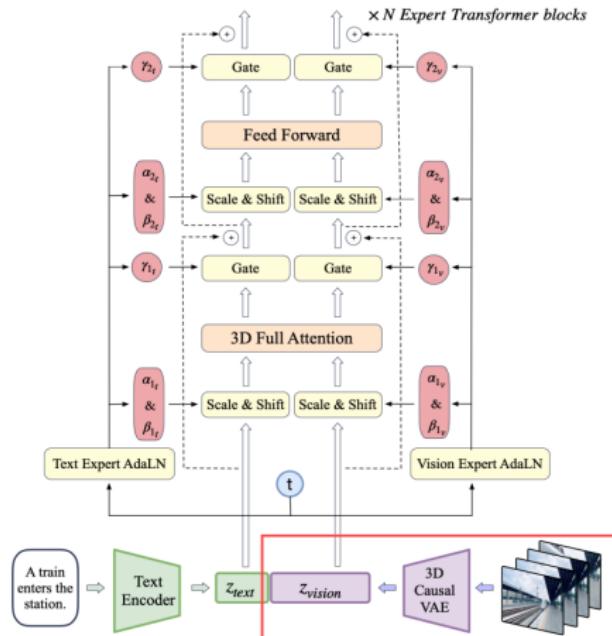
5. SOTA Methods

Here, we introduce the SOTA methods that are widely adopted these days, and explain how their underlying architecture is correlated to the ones we have introduced. We will briefly introduce **5.1 CogVideoX**, **5.2 Sora**, **5.3 Wan-video**, and how they are correlated to our previous introductions.

5. SOTA Methods

5.1 CogVideoX²⁰

CogVideoX is performed in the 3D temporal causal VAE latent space, as we have described in Section 2.2, with a Transformer framework

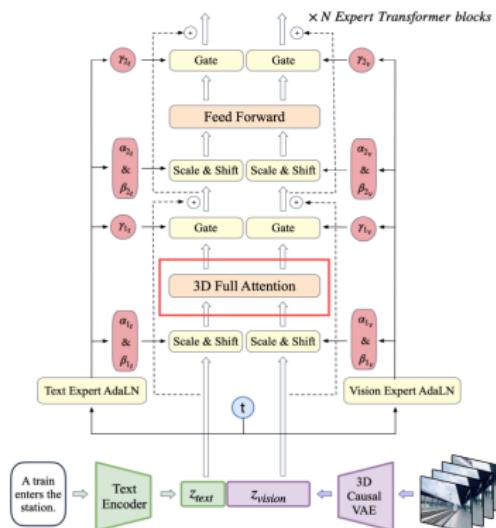


²⁰CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer

5. SOTA Methods

5.1 CogVideoX

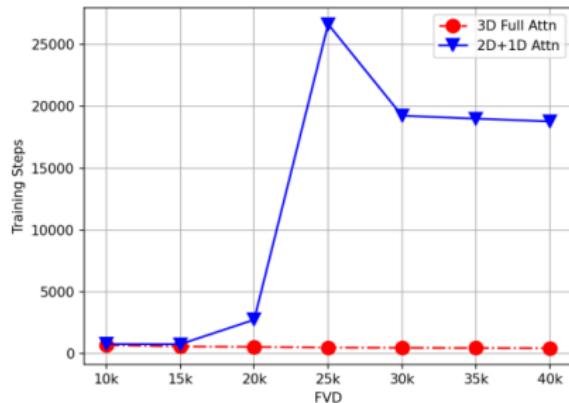
The embedded text and video frames are then sent into the transformer blocks. Although the decomposed attention in Section 1.2.2 can significantly reduce computation burden, the author states that 3D full attention can better capture the motion between frames than the decomposed version.



5. SOTA Methods

5.1 CogVideoX

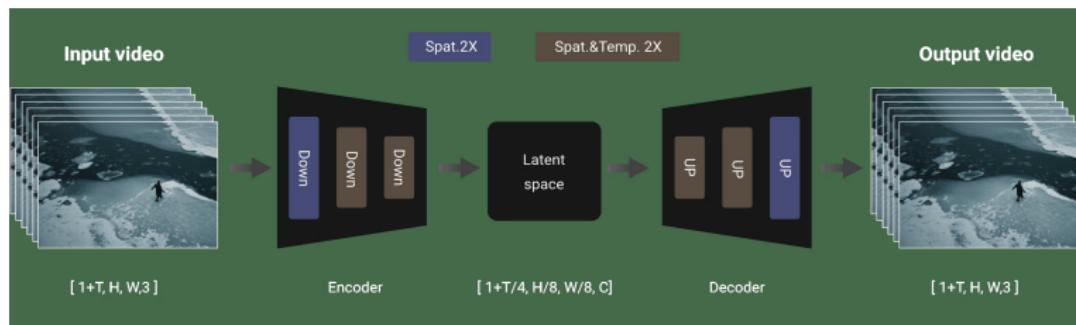
Further ablation studies also show that the 3D full attention significantly increase the stability of the model training compared to the divided time-space attention, as stated in Section 1.2.2.



5. SOTA Methods

5.2 Wan-video

Wan-video²¹ is a DiT-based framework in VAE latent space. Similar to the strategy in Section 5.1, it also implement a 3D temporal causal VAE for video compression.

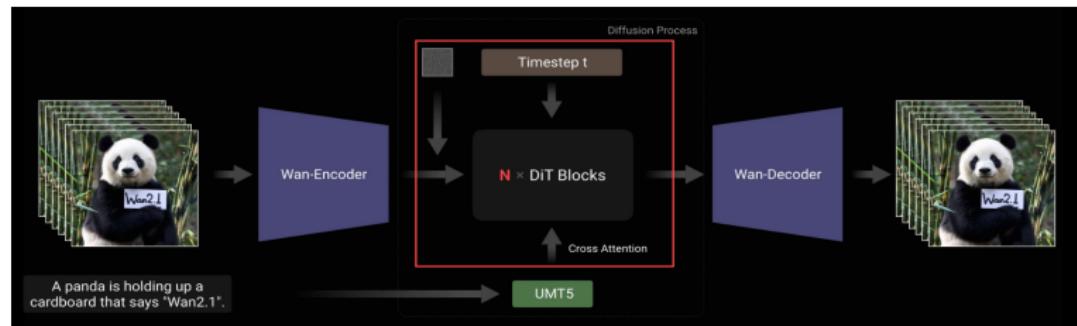


²¹Wan: Open and Advanced Large-Scale Video Generative Models

5. SOTA Methods

5.2 Wan-video

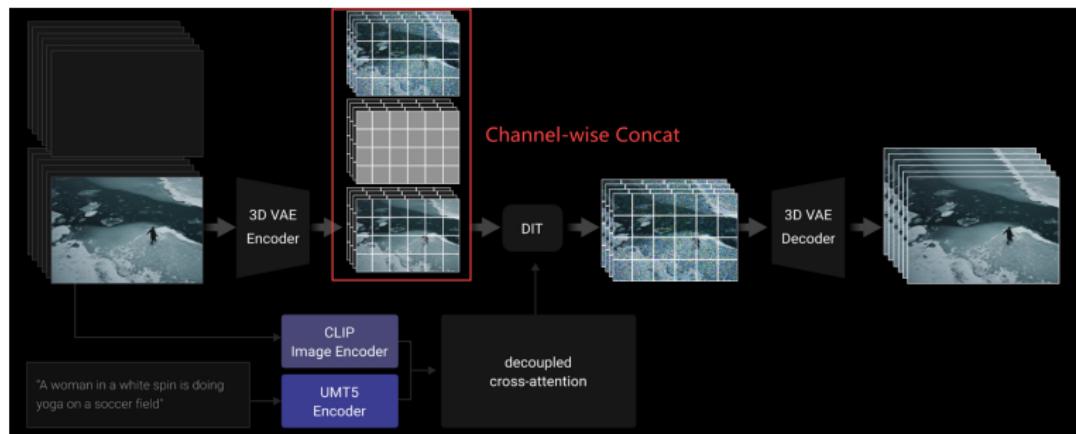
The framework for Wan-video in T2V task is DiT, which we have introduced in Section 1.3, but the approach to handle temporal information is not specified in the technical report.



5. SOTA Methods

5.2 Wan-video

As for the I2V task, Wan-video adopt the first kind approach (i.e. via concatenating the binary mask, input frame with the initial noise) we have introduced in Section 4.1.



5. SOTA Methods

5.3 Sora²²

Sora is performed also in a VAE latent space, but the specific type of VAE is not stated in their technical report.

As we have stated in Section 1.3, Sora adopt a similar pipeline with Latte, which employs **3D DiT** as the backbone in the latent space. Such an architecture yields better scalability, and can **accept arbitrary sized input** image, compared to 3D UNet-based method that needs cropping and resizing of the input frames.

²²OpenAI, “Sora: Creating video from text.” <https://openai.com/sora>, 2024.

5. SOTA Methods

5.3 Sora

Training on data in their native **sizes and aspect ratios** brings better framing in the generated videos.



(a) Training on videos that are cropped to squares leads to unnatural compositions and framing.



(b) Training in native sizes improves framing.

In the figure above, (b) provides a full video of the text description, while the key concepts in (a) are partly cropped since the training data are also cropped.

5. SOTA Methods

5.3 Sora

The most naive strategy is to support dynamic sizes across different batches by grouping samples into pre-defined "buckets" with minimum resizing or cropping. Within each batch, the resolution and aspect ratio are fixed.

Another strategy is to pack multiple patches from different images (latents) in a single sequence as shown below:

