10-423/10-623 Generative AI

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Video Generation and Understanding

Matt Gormley & Pat Virtue
Lecture 25
Apr. 21, 2025

Reminders

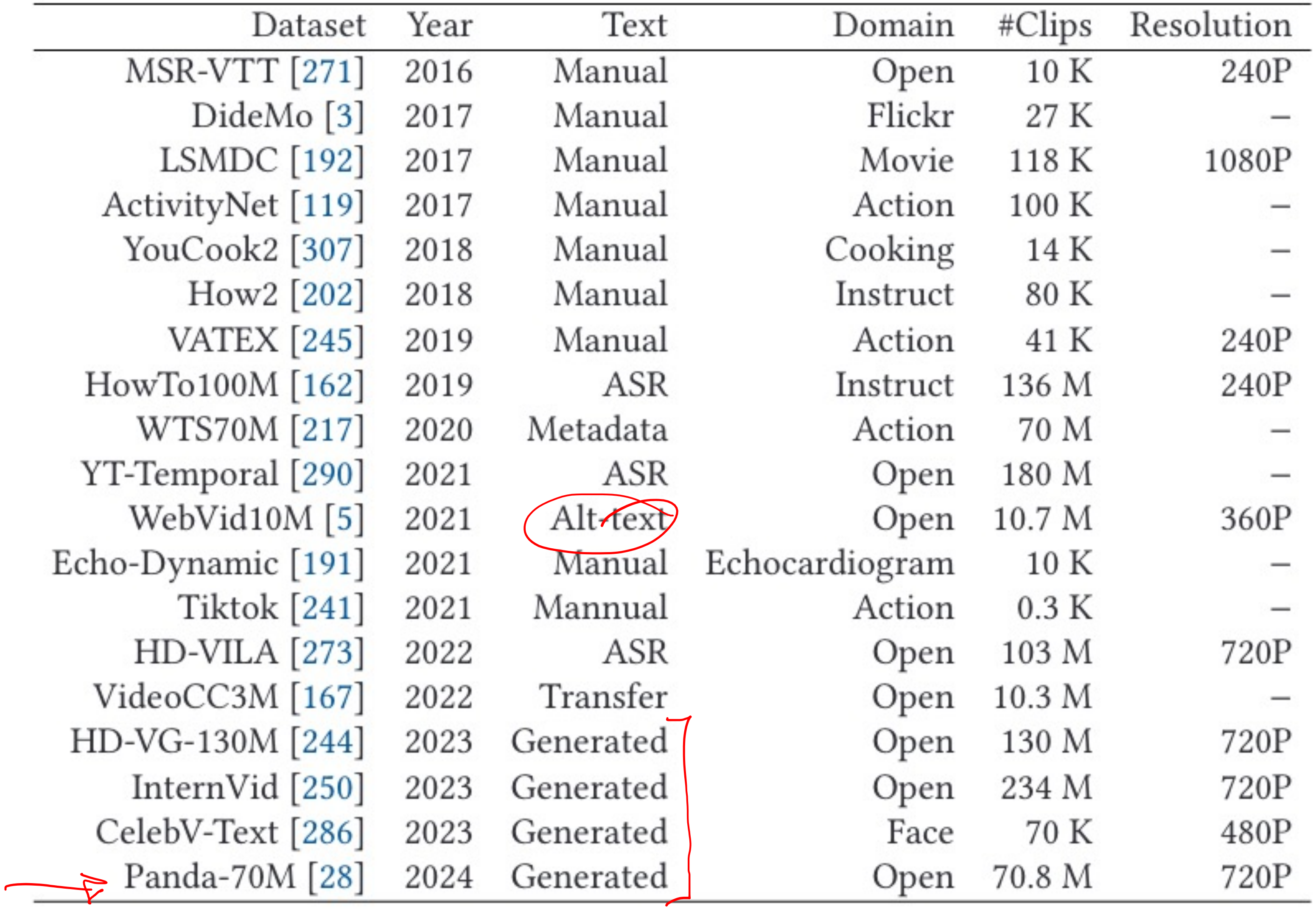
- HW623
  - Only for students registered in 10-623
  - Due: Mon, Apr-21 at 11:59pm
- Project “Poster”
  - Upload Due: Sun, Apr-27 at 11:59pm
  - Presentations: Tue, Apr-29 at 8:30am-11:30am
- Project Final Report
  - Due: Thu, May 1 at 11:59pm
- Project Code Upload
  - Due: Thu, May 1 at 11:59pm

Video Generation and Understanding

- Outline
  - video generation (video diffusion models)
    - 3D Unet (2016)
    - VViT & Spatio-Temporal Attention (2021)
    - **Video Diffusion Model** (2022)
    - **Video Latent Diffusion Model** (2023)
    - Diffusion Transformer (2023)
    - **Sora** (2024)
  - video understanding (visual language models)
    - **Large World Model** (generate/understand text+image+video)

VIDEO DIFFUSION MODELS

Datasets for Video Models

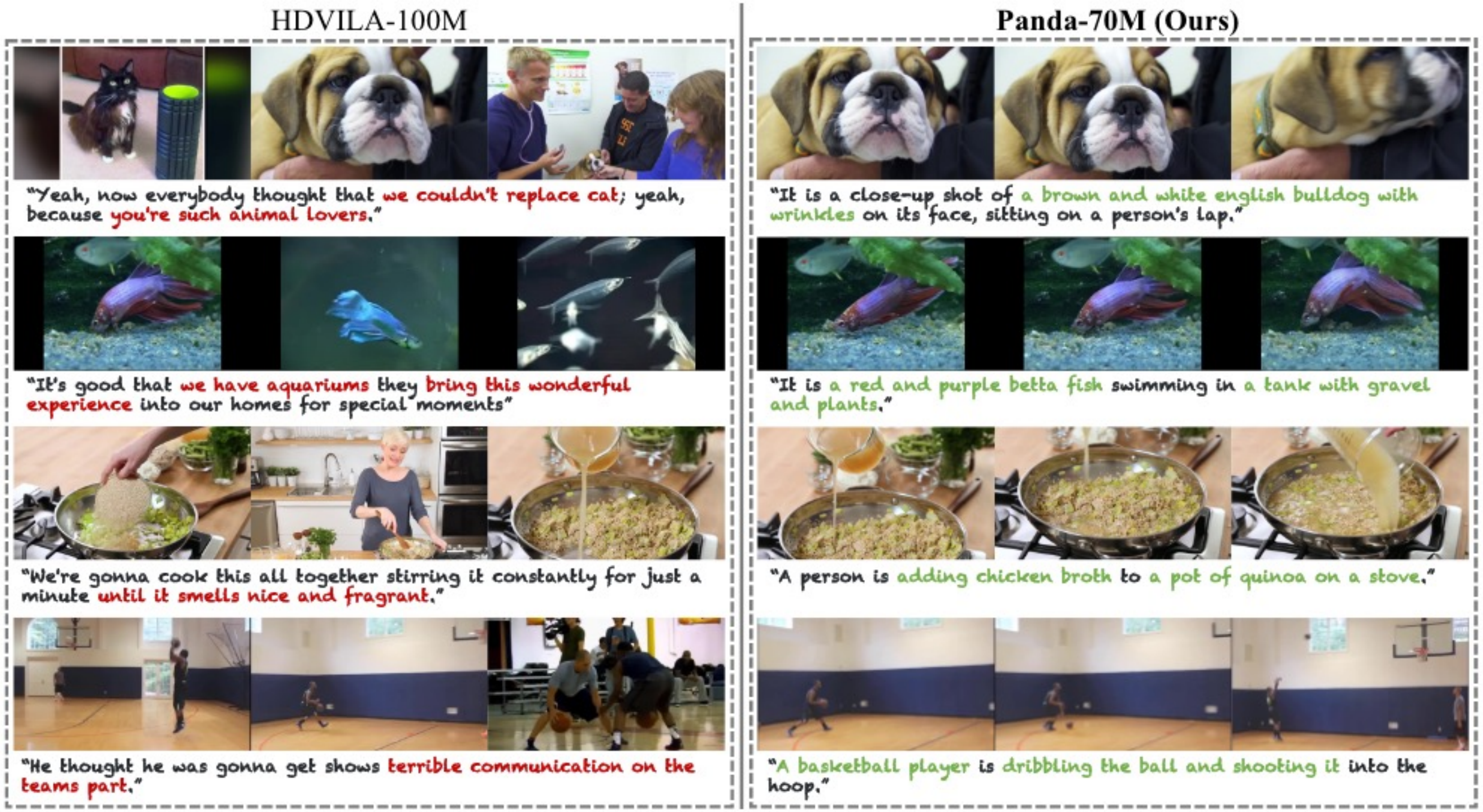


Dataset	Year	Text	Domain	#Clips	Resolution
MSR-VTT [271]	2016	Manual	Open	10 K	240P
DideMo [3]	2017	Manual	Flickr	27 K	–
LSMDC [192]	2017	Manual	Movie	118 K	1080P
ActivityNet [119]	2017	Manual	Action	100 K	–
YouCook2 [307]	2018	Manual	Cooking	14 K	–
How2 [202]	2018	Manual	Instruct	80 K	–
VATEX [245]	2019	Manual	Action	41 K	240P
HowTo100M [162]	2019	ASR	Instruct	136 M	240P
WTS70M [217]	2020	Metadata	Action	70 M	–
YT-Temporal [290]	2021	ASR	Open	180 M	–
WebVid10M [5]	2021	Alt-text	Open	10.7 M	360P
Echo-Dynamic [191]	2021	Manual	Echocardiogram	10 K	–
Tiktok [241]	2021	Mannual	Action	0.3 K	–
HD-VILA [273]	2022	ASR	Open	103 M	720P
VideoCC3M [167]	2022	Transfer	Open	10.3 M	–
HD-VG-130M [244]	2023	Generated	Open	130 M	720P
InternVid [250]	2023	Generated	Open	234 M	720P
CelebV-Text [286]	2023	Generated	Face	70 K	480P
Panda-70M [28]	2024	Generated	Open	70.8 M	720P

Figure from https://dl.acm.org/doi/10.1145/3696415

Datasets for Video Models

- The largest datasets of videos with captions use a model to generate the captions
- The quality and style of the captions can vary wildly depending on which model is used



"Yeah, now everybody thought that we couldn't replace cat; yeah, because you're such animal lovers."

"It is a close-up shot of a brown and white english bulldog with wrinkles on its face, sitting on a person's lap."

"It's good that we have aquariums they bring this wonderful experience into our homes for special moments"

"It is a red and purple betta fish swimming in a tank with gravel and plants."

"We're gonna cook this all together stirring it constantly for just a minute until it smells nice and fragrant."

"A person is adding chicken broth to a pot of quinoa on a stove."

"He thought he was gonna get shows terrible communication on the teams part."

"A basketball player is dribbling the ball and shooting it into the hoop."

Figure 1. **Comparison of Panda-70M to the existing large-scale video-language datasets.** We introduce Panda-70M, a large-scale video dataset with captions that are annotated by multiple cross-modality vision-language models. Compared to text annotations in existing dataset [80], captions in Panda-70M more precisely describe the main object and action in videos (highlighted in green). Besides, videos in Panda-70M are semantically coherent, high-resolution, and free from watermarks. More samples can be found in Appendix E.

Panda-70M Examples: https://snap-research.github.io/Panda-70M/

Figure from http://arxiv.org/abs/2402.19479

Video Diffusion Model

- Architecture = 3D UNet + spatial-temporal attention
- Relative positional embeddings across time axis

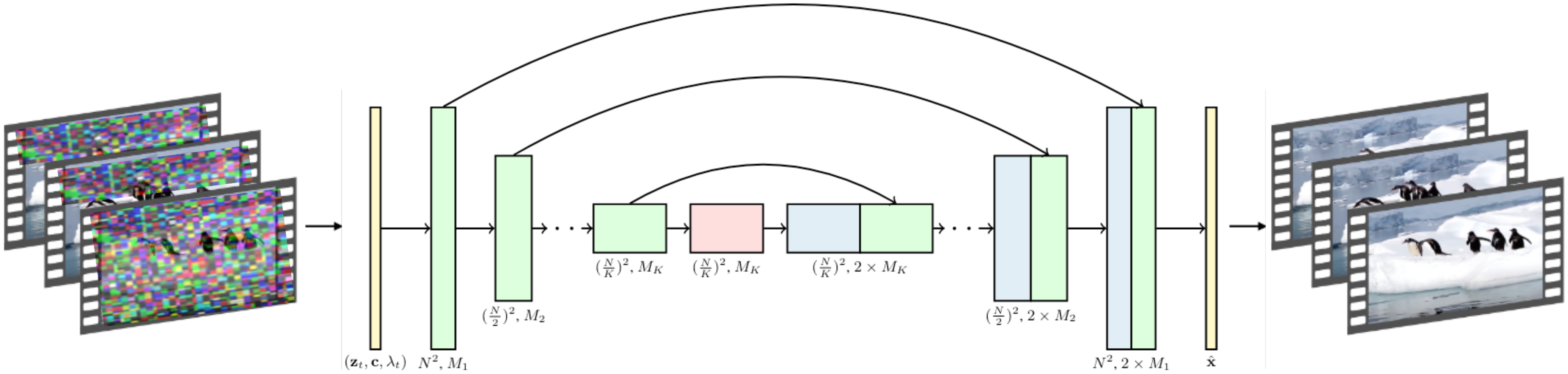
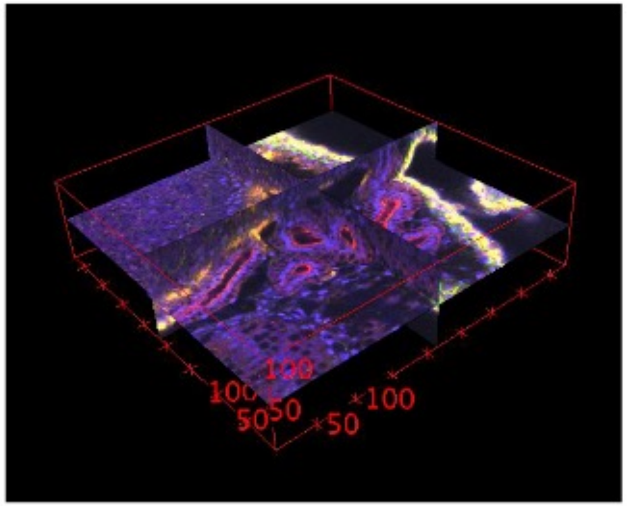


Figure 1: The 3D U-Net architecture for $\hat{\mathbf{x}}_\theta$ in the diffusion model. Each block represents a 4D tensor with axes labeled as frames × height × width × channels, processed in a space-time factorized manner as described in Section 3. The input is a noisy video $\mathbf{z}_t$, conditioning $\mathbf{c}$, and the log SNR λ_t. The downsampling/upsampling blocks adjust the spatial input resolution height × width by a factor of 2 through each of the K blocks. The channel counts are specified using channel multipliers M_1, M_2, ..., M_K, and the upsampling pass has concatenation skip connections to the downsampling pass.

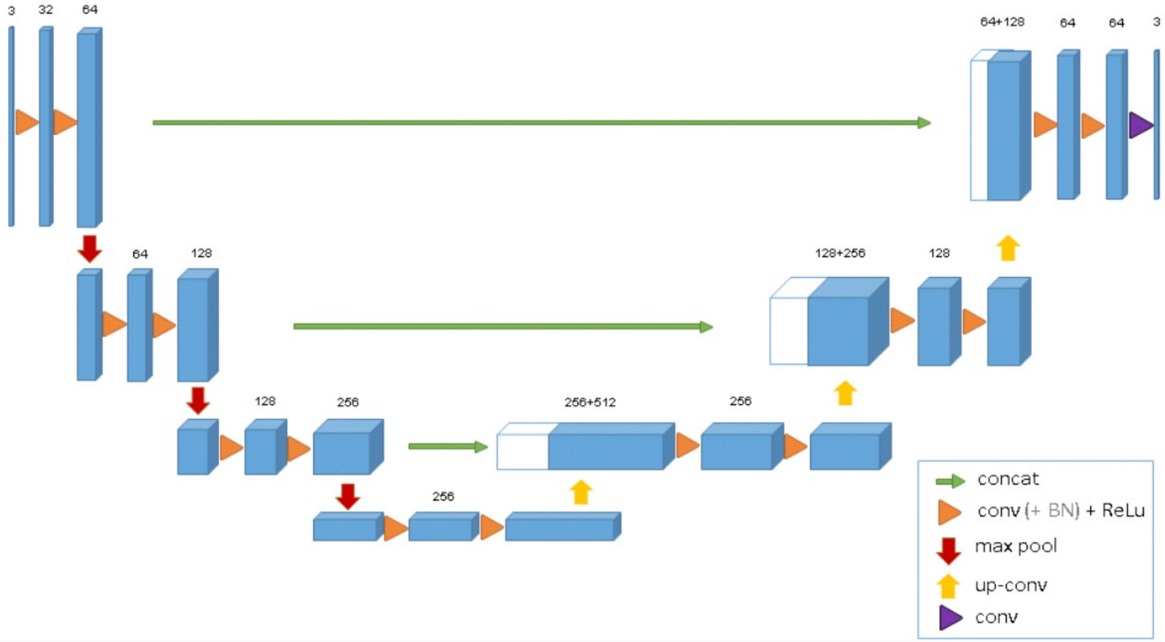
Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

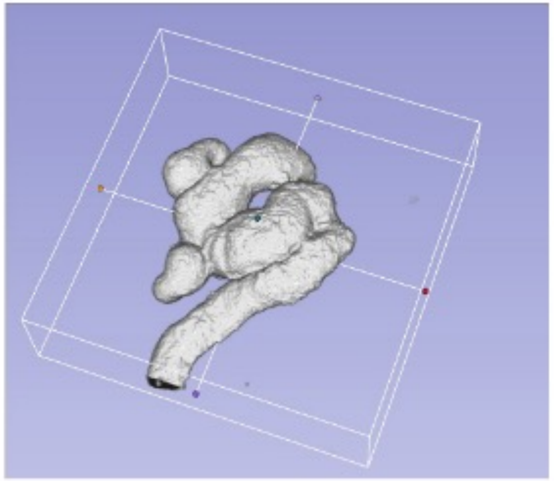
3D UNet

- Suppose you want to do image segmentation on 3D images (e.g. high-resolution, 3D imaging of a *Xenopus* kidney)
- The 3D UNet model is almost identical to the standard UNet except that it replaces 2D convolution (height, width, channel) with 3D convolution (height, width, **depth**, channel)



raw image





dense segmentation

Figure from http://arxiv.org/abs/1606.06

Video Vision Transformer (VViT)

- The Video ViT takes a series of image frames from a video as input
- A standard ViT model for images would treat each frame as independent (i.e. only has spatial attention across the [w,h] axes)
- The Video ViT instead includes two types of attention: spatial attention and temporal attention

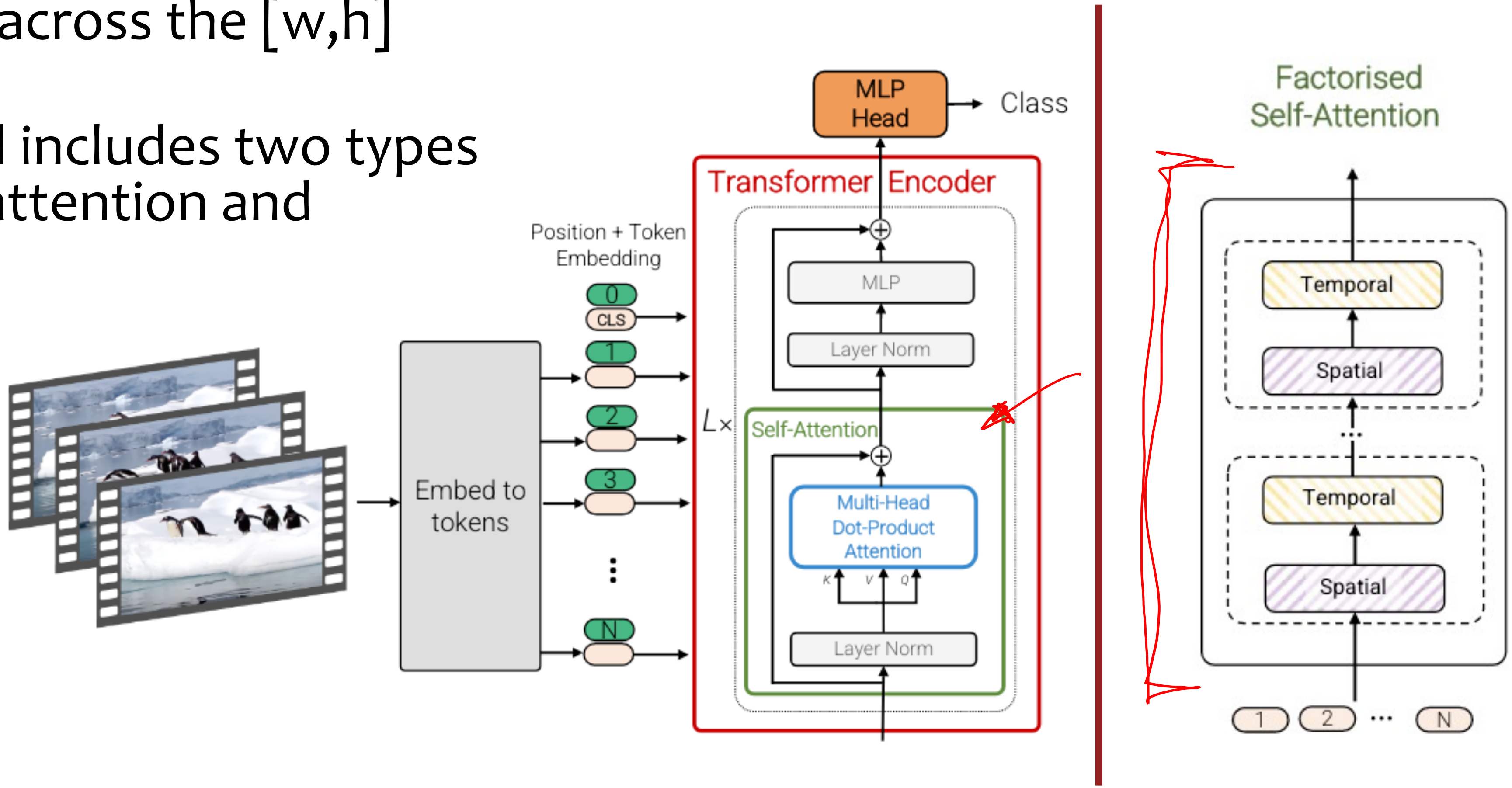
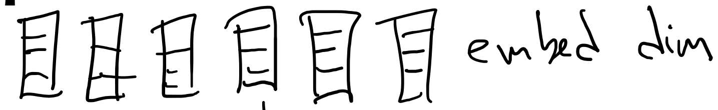


Figure from http://arxiv.org/abs/2103.15691

Factorized Spatial-Temporal Attention



timestep

VViT alternate between the two types:

Spatial attention:

1. reshape: b t h w c -> (b t) (h w) c
2. multi-headed attention

batch timestep embedding dim

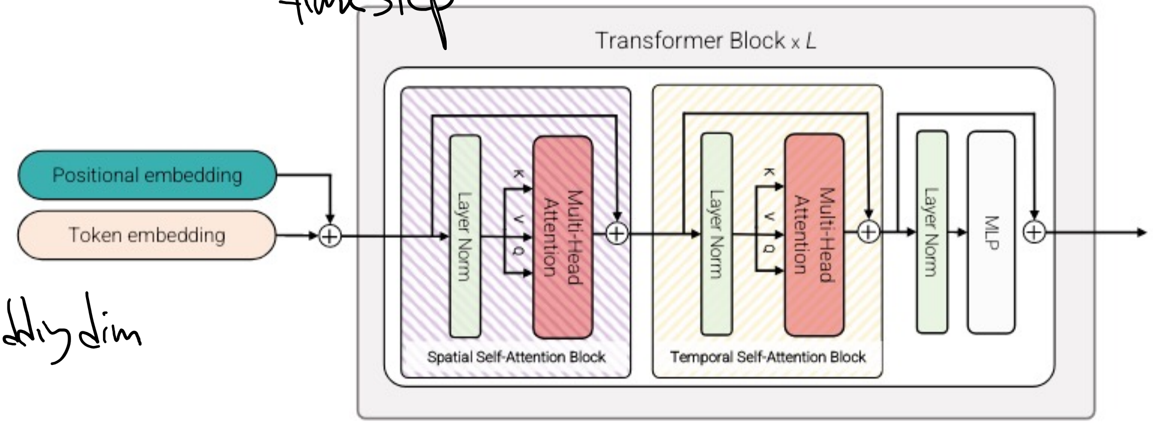
3. reshape: (b t) (h w) c -> b t h w c

frames

Temporal attention:

1. reshape: b t h w c -> (b h w) t c
2. multi-headed attention
3. reshape: (b h w) t c -> b t c h w

b



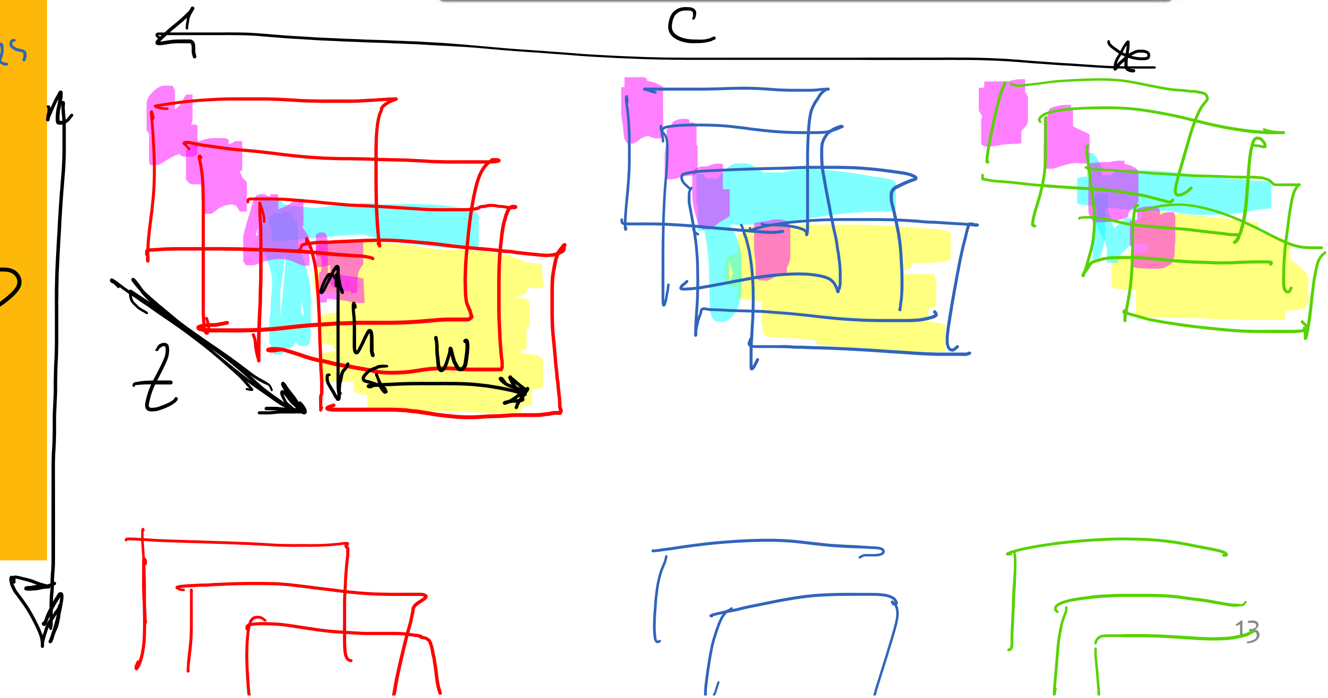


Figure from http://arxiv.org/abs/2103.15691

Video Diffusion Model

- Architecture = 3D UNet + spatial-temporal attention
- Relative positional embeddings across time axis

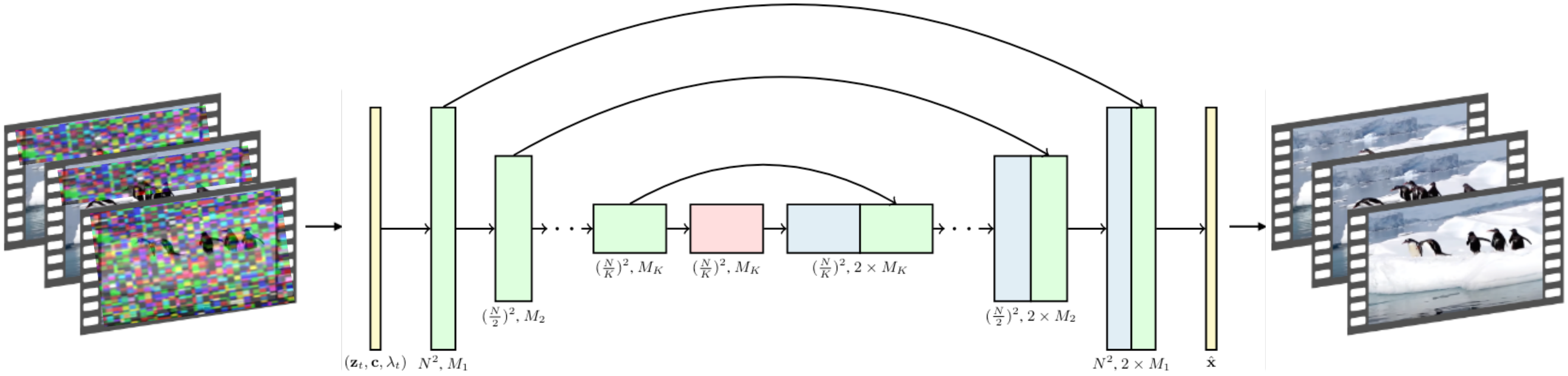


Figure 1: The 3D U-Net architecture for $\hat{\mathbf{x}}_\theta$ in the diffusion model. Each block represents a 4D tensor with axes labeled as frames × height × width × channels, processed in a space-time factorized manner as described in Section 3. The input is a noisy video $\mathbf{z}_t$, conditioning $\mathbf{c}$, and the log SNR λ_t. The downsampling/upsampling blocks adjust the spatial input resolution height × width by a factor of 2 through each of the K blocks. The channel counts are specified using channel multipliers M_1, M_2, ..., M_K, and the upsampling pass has concatenation skip connections to the downsampling pass.

Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Diffusion Model

Figure from http://arxiv.org/abs/2304.08818

- The model can be **jointly** trained on **images and video**
- When trained on images, the temporal attention is **masked** so that all the attention mass is placed on the current batch element
- Such training improves performance on video generation

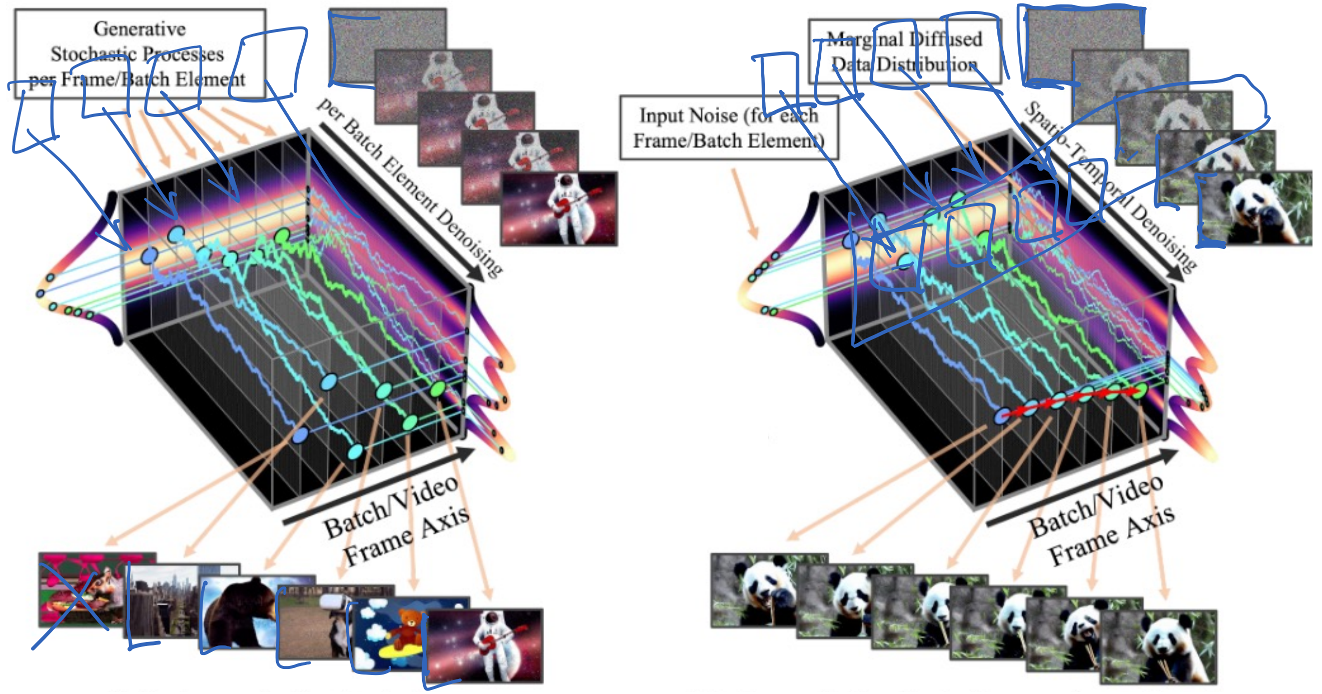

Generative Stochastic Processes per Frame/Batch Element
Per Batch Element Denoising
Batch/Video Frame Axis
Input Noise (for each Frame/Batch Element)
Marginal Diffused Data Distribution
Spatio-Temporal Denoising
Batch/Video Frame Axis

Image frames	FVD↓	FID-avg↓	IS-avg↑	FID-first↓	IS-first↑
0	202.28/205.42	37.52/37.40	7.91/7.58	41.14/40.87	9.23/8.74
4	68.11/70.74	18.62/18.42	9.02/8.53	22.54/22.19	10.58/9.91
8	57.84/60.72	15.57/15.44	9.32/8.82	19.25/18.98	10.81/10.12

Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Diffusion Model

- **Classifier-free Guidance**

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, \overset{\text{text}}{\mathbf{c}}) = (1+w)\underbrace{\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \overset{\text{prompt}}{\mathbf{c}})}_{\text{3DUNet}} - w\underbrace{\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)}_{\text{3DUNet}},$$

- Suppose we want to sample from a conditional distribution: $p_\theta(\mathbf{x}^{\mathrm{b}}|\mathbf{x}^{\mathrm{a}})$.
  - $\mathbf{x}^{\mathrm{a}}$ could be the first 16 frames, and now we want to generate the next 16 frames $\mathbf{x}^{\mathrm{b}}$
  - or $\mathbf{x}^{\mathrm{a}}$ could be a low frame rate video and $\mathbf{x}^{\mathrm{b}}$ are the frames in between to increase framerate
- **Reconstruction Guided Sampling**
  - key idea: guide the sample “based on the model’s reconstruction of the conditioning data”

$$\tilde{\mathbf{x}}^b_\theta(\mathbf{z}_t) = \underbrace{\hat{\mathbf{x}}^b_\theta(\mathbf{z}_t)}_{} - \frac{w_r \alpha_t}{2} \nabla_{\mathbf{z}^b_t} \|\mathbf{x}^a - \underbrace{\hat{\mathbf{x}}^a_\theta(\mathbf{z}_t)}_{}\|^2_2$$

Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Diffusion Model

- **Reconstruction Guided Sampling**

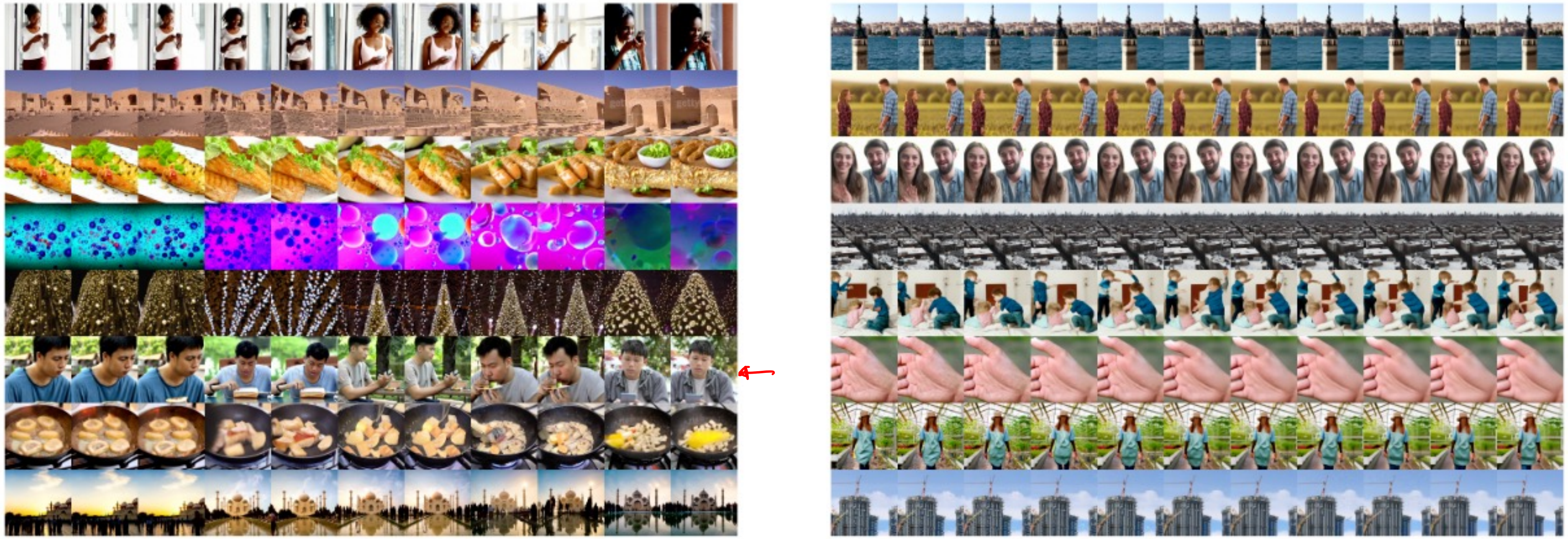


Figure 4: Comparing the replacement method (left) vs the reconstruction guidance method (right) for conditioning for block-autoregressive generation of 64 frames from a 16 frame model. Video frames are displayed over time from left to right; each row is an independent sample. The replacement method suffers from a lack of temporal coherence, unlike the reconstruction guidance method.

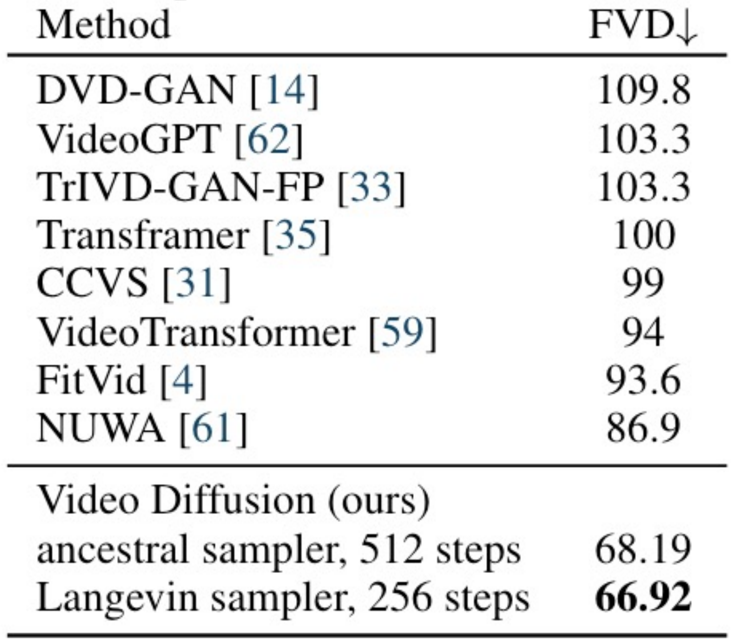
Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Diffusion Model

Results

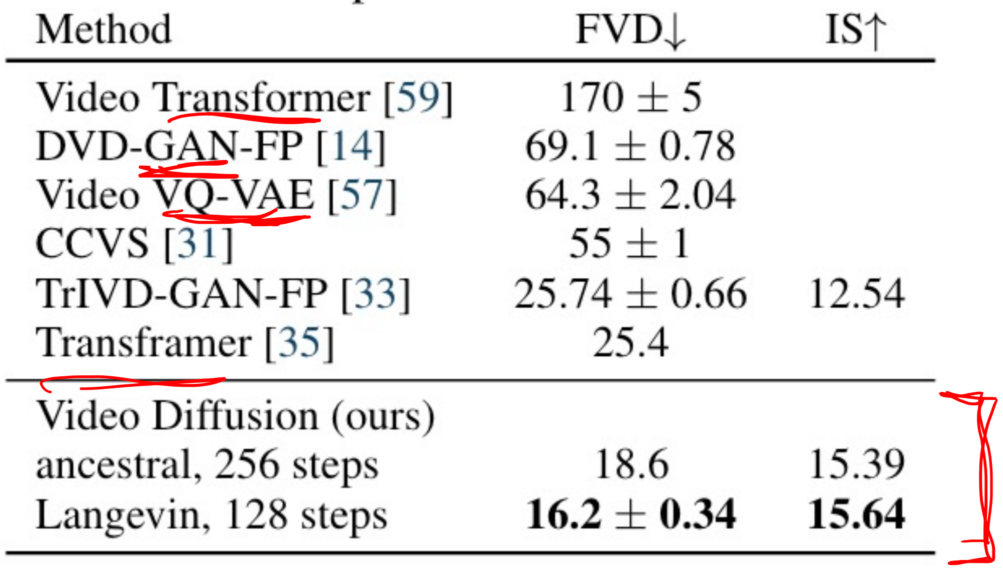
- In video prediction, we are given the beginning of a video and we see how well the model can complete it

Table 2: Video prediction on BAIR Robot Pushing.



Method	FVD↓
DVD-GAN [14]	109.8
VideoGPT [62]	103.3
TrIVD-GAN-FP [33]	103.3
Transframer [35]	100
CCVS [31]	99
VideoTransformer [59]	94
FitVid [4]	93.6
NUWA [61]	86.9
Video Diffusion (ours)	
ancestral sampler, 512 steps	68.19
Langevin sampler, 256 steps	**66.92**

Table 3: Video prediction on Kinetics-600.



Method	FVD↓	IS↑
Video Transformer [59]	170 ± 5	
DVD-GAN-FP [14]	69.1 ± 0.78	
Video VQ-VAE [57]	64.3 ± 2.04	
CCVS [31]	55 ± 1	
TrIVD-GAN-FP [33]	25.74 ± 0.66	12.54
Transframer [35]	25.4	
Video Diffusion (ours)		
ancestral, 256 steps	18.6	15.39
Langevin, 128 steps	**16.2 ± 0.34**	**15.64**

Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Diffusion Model

Results

- In video prediction, we are given the beginning of a video and we see how well the model can complete it

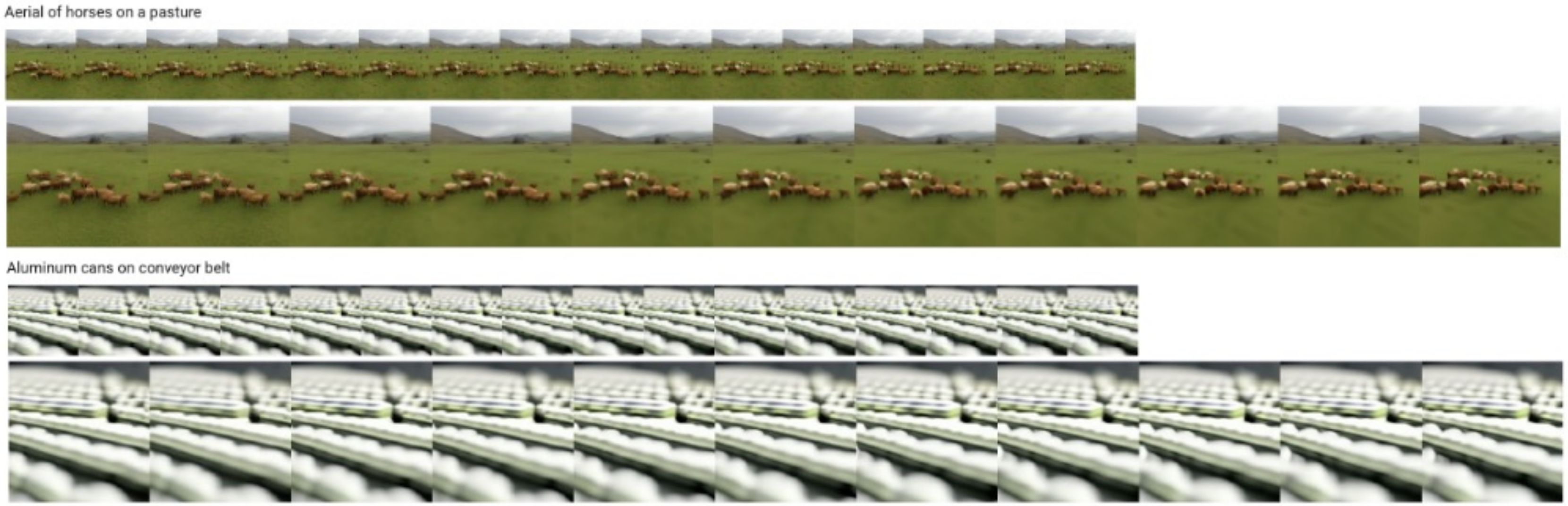


Figure 2: Text-conditioned video samples from a cascade of two models. First samples are generated from a 16x64x64 frameskip 4 model. Then those samples are treated as ground truth for simultaneous super-resolution and autoregressive extension to 64x128x128 using a 9x128x128 frameskip 1 model. Both models are conditioned on the text prompt. In this figure, the text prompt, low resolution frames, and high resolution frames are visualized in sequence. See Fig. 5 for more samples.

Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Diffusion Model

- Demo: https://video-diffusion.github.io/

Figure from https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html

Video Latent Diffusion Model (VLDM)

- The VLDM model combines two pieces:
  - Encoder/Decoder trained to reconstruct convert videos down to a latent representation and then reconstruct them
  - A video diffusion model trained to work in the latent space

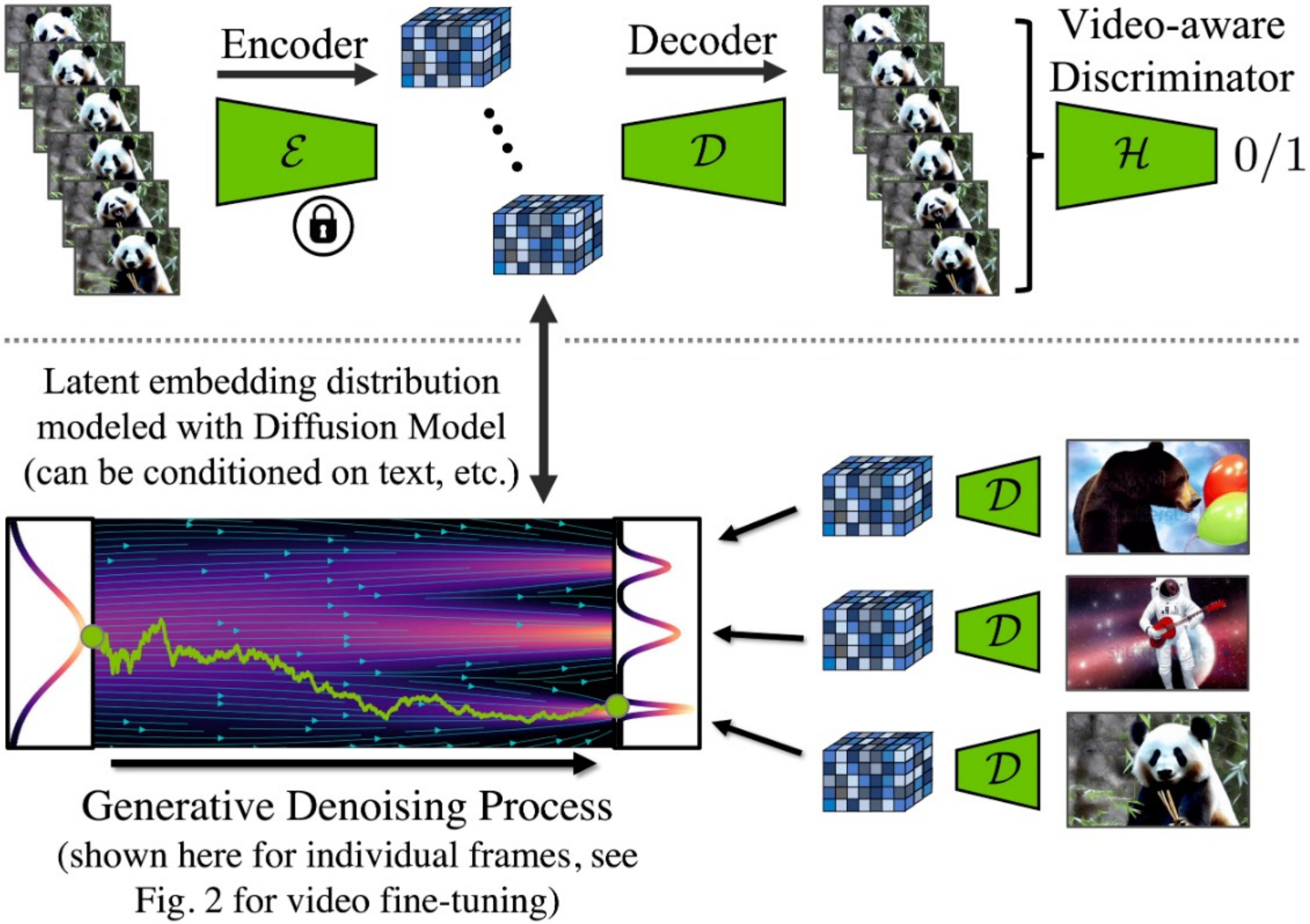

Encoder
$\mathcal{E}$
Decoder
$\mathcal{D}$
Video-aware
Discriminator
$\mathcal{H}$
0/1
Latent embedding distribution
modeled with Diffusion Model
(can be conditioned on text, etc.)
$\mathcal{D}$
$\mathcal{D}$
$\mathcal{D}$
Generative Denoising Process
(shown here for individual frames, see
Fig. 2 for video fine-tuning)

Figure from http://arxiv.org/abs/2304.08818

Video Latent Diffusion Model (VLDM)

- The VLDM model combines two pieces:
  - Encoder/Decoder trained to reconstruct convert videos down to a latent representation and then reconstruct them
  - A video diffusion model trained to work in the latent space

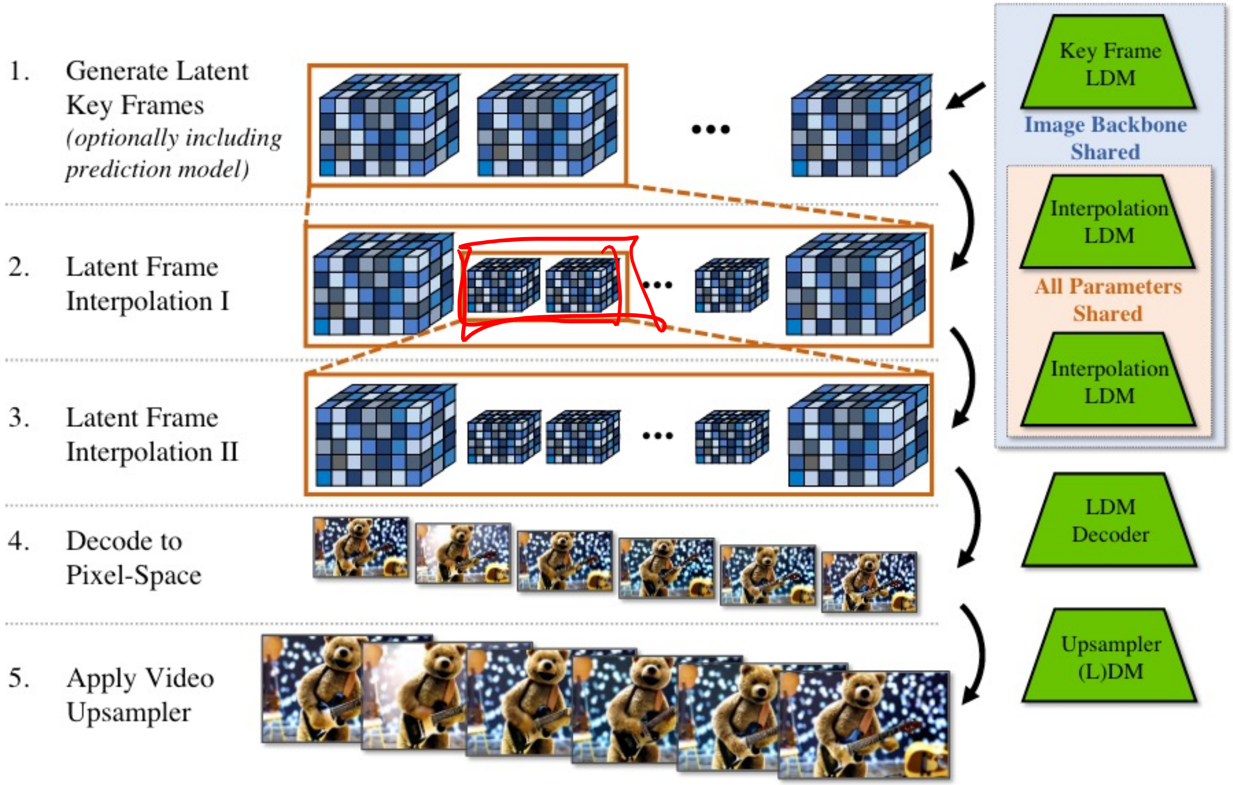


Figure 5. **Video LDM Stack.** We first generate sparse key frames. Then we temporally interpolate in two steps with the same interpolation model to achieve high frame rates. These operations are all based on latent diffusion models (LDMs) that share the same image backbone. Finally, the latent video is decoded to pixel space and optionally a video upsampler diffusion model is applied.

Figure from http://arxiv.org/abs/2304.08818

Video Latent Diffusion Model (VLDM)

- Demo: https://research.nvidia.com/labs/toronto-ai/VideoLDM/

Figure from http://arxiv.org/abs/2304.08818

Diffusion Transformer

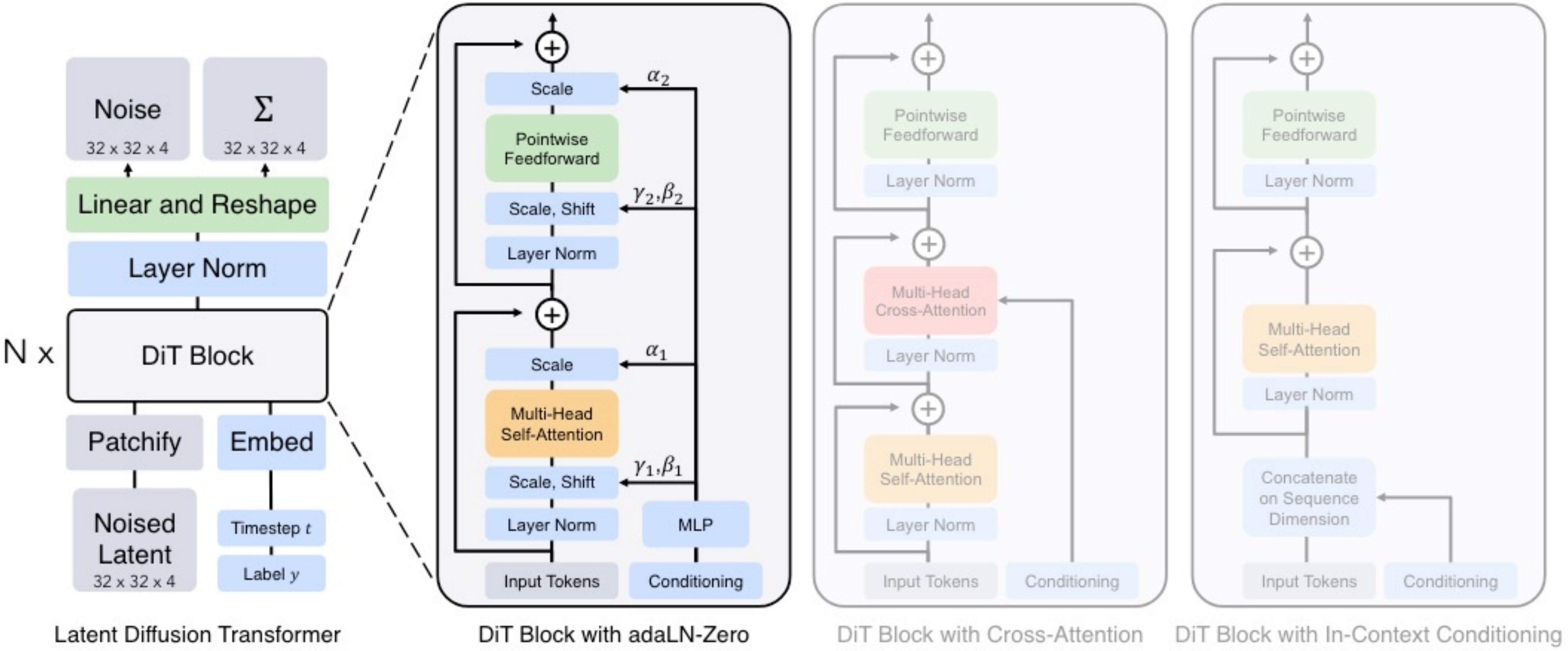


Figure 3. **The Diffusion Transformer (DiT) architecture.** *Left:* We train conditional latent DiT models. The input latent is decomposed into patches and processed by several DiT blocks. *Right:* Details of our DiT blocks. We experiment with variants of standard transformer blocks that incorporate conditioning via adaptive layer norm, cross-attention and extra input tokens. Adaptive layer norm works best.

Figure from http://arxiv.org/abs/2212.09748

Diffusion Transformer

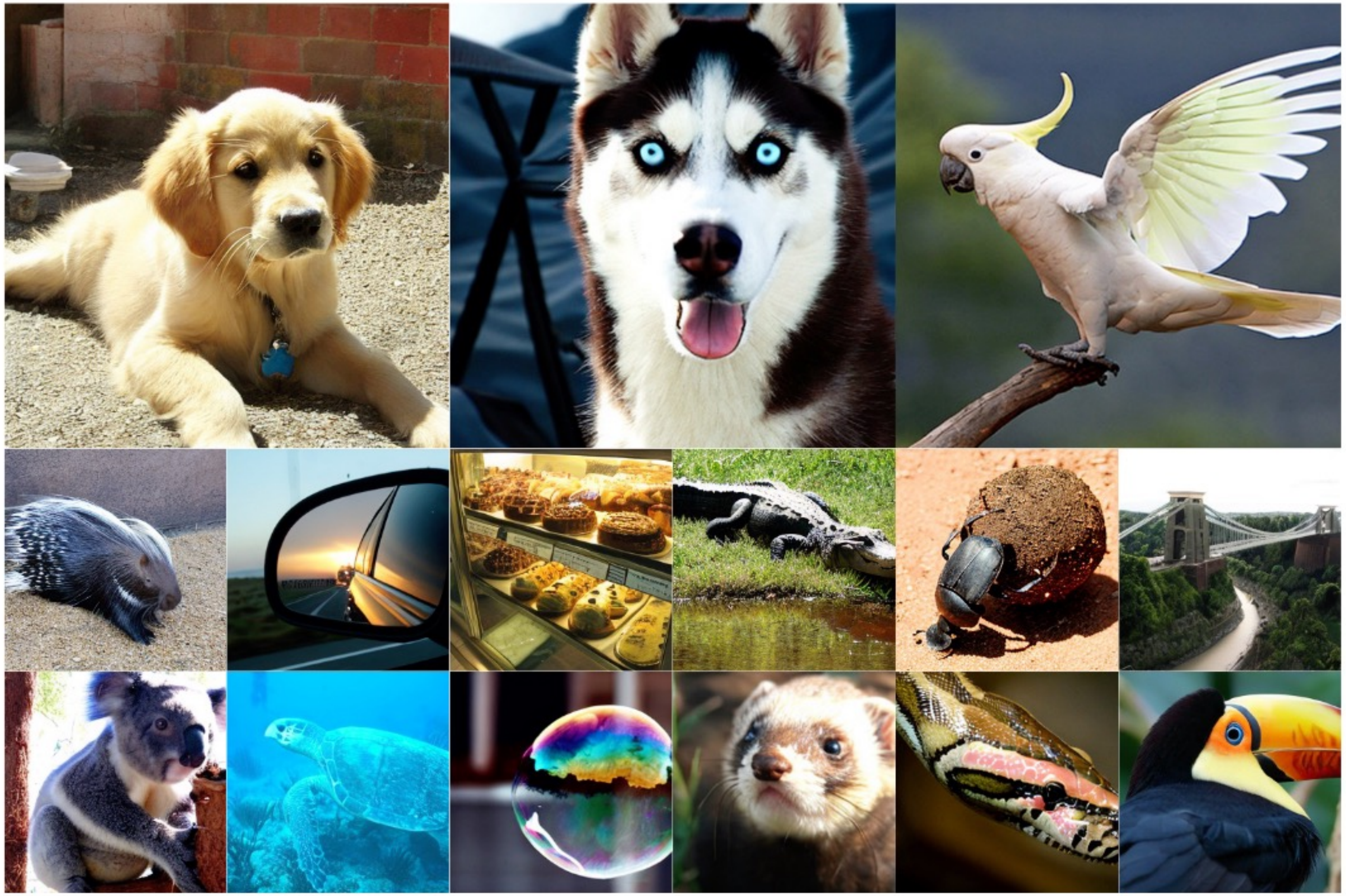
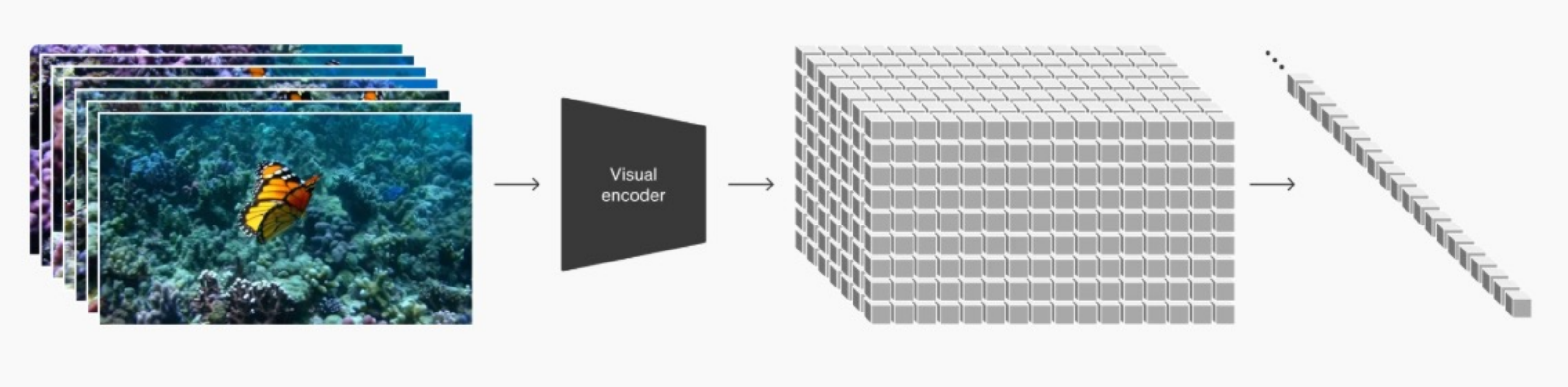


Figure 1. **Diffusion models with transformer backbones achieve state-of-the-art image quality.** We show selected samples from two of our class-conditional DiT-XL/2 models trained on ImageNet at 512×512 and 256×256 resolution, respectively.

Figure from http://arxiv.org/abs/2212.09748

Sora



Sora uses a Diffusion Transformer backbone trained on images and videos

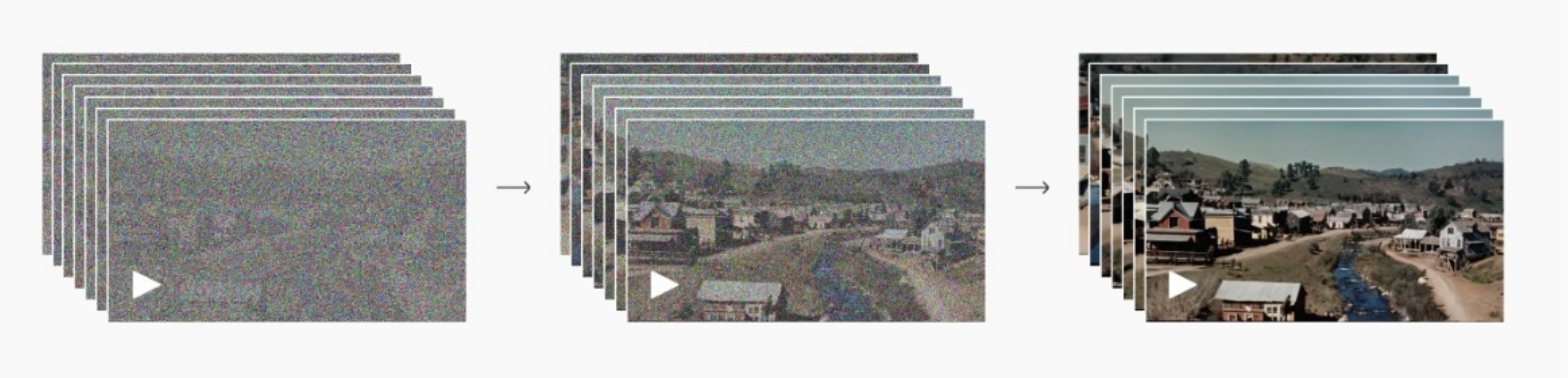
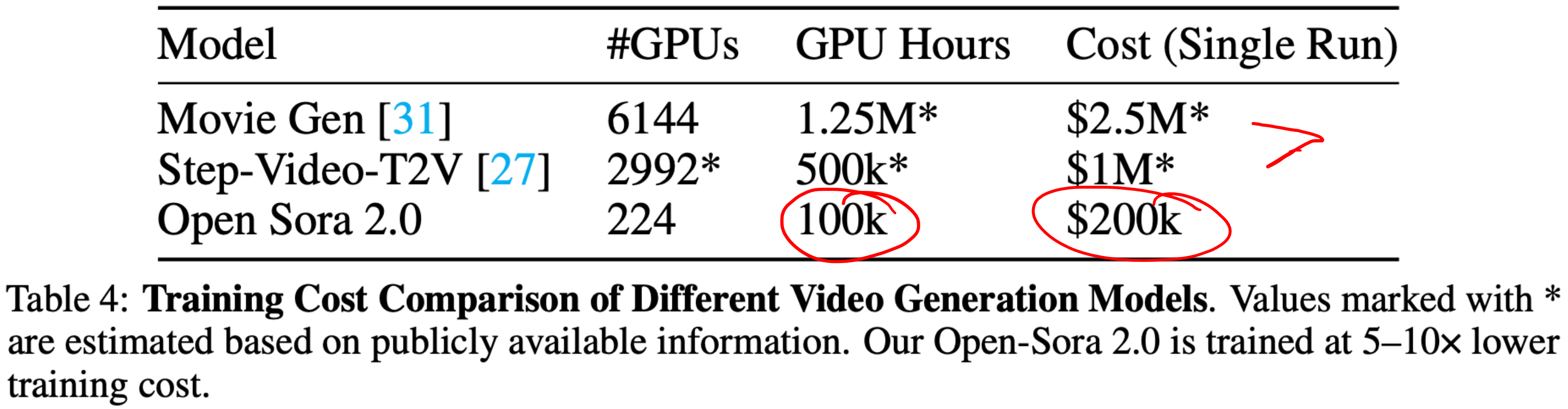


Figure from https://openai.com/index/video-generation-models-as-world-simulators/

OpenSora

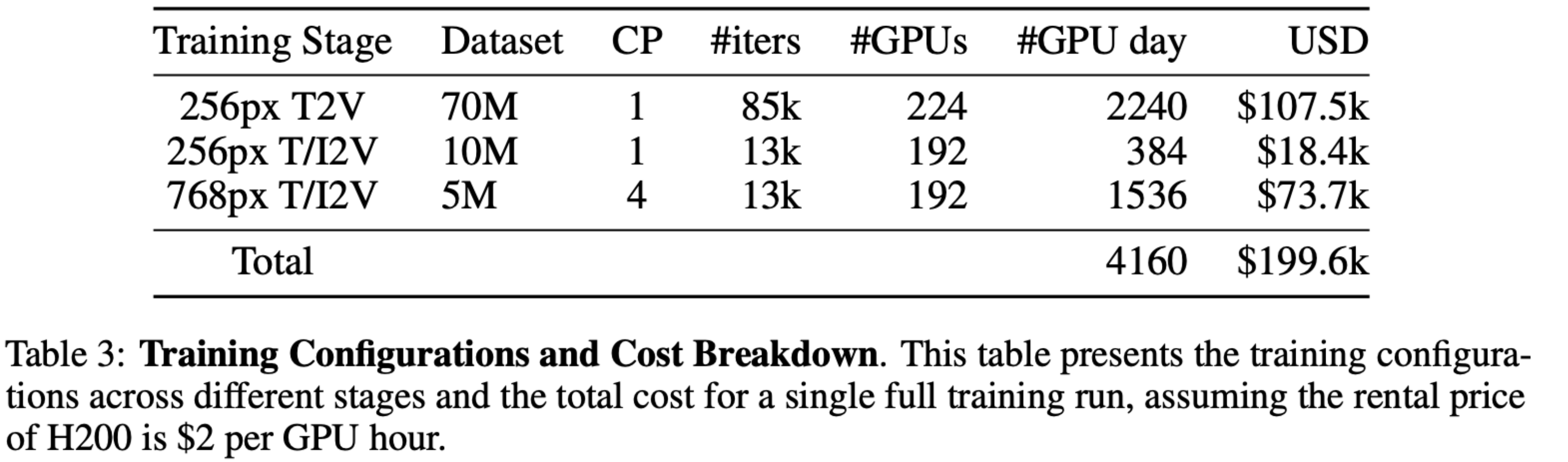
- Training costs are trending downward as we improve the models / datasets



Model	#GPUs	GPU Hours	Cost (Single Run)
Movie Gen [31]	6144	1.25M*	$2.5M*
Step-Video-T2V [27]	2992*	500k*	$1M*
Open Sora 2.0	224	100k	$200k

Table 4: **Training Cost Comparison of Different Video Generation Models**. Values marked with * are estimated based on publicly available information. Our Open-Sora 2.0 is trained at 5–10× lower training cost.

- OpenSora employs a three stage training approach:



Training Stage	Dataset	CP	#iters	#GPUs	#GPU day	USD
256px T2V	70M	1	85k	224	2240	$107.5k
256px T/I2V	10M	1	13k	192	384	$18.4k
768px T/I2V	5M	4	13k	192	1536	$73.7k
Total					4160	$199.6k

Table 3: **Training Configurations and Cost Breakdown**. This table presents the training configurations across different stages and the total cost for a single full training run, assuming the rental price of H200 is $2 per GPU hour.

https://arxiv.org/pdf/2503.09642v1

OpenSora

How do we evaluate?

- Human evaluation
- Benchmarks + Metrics

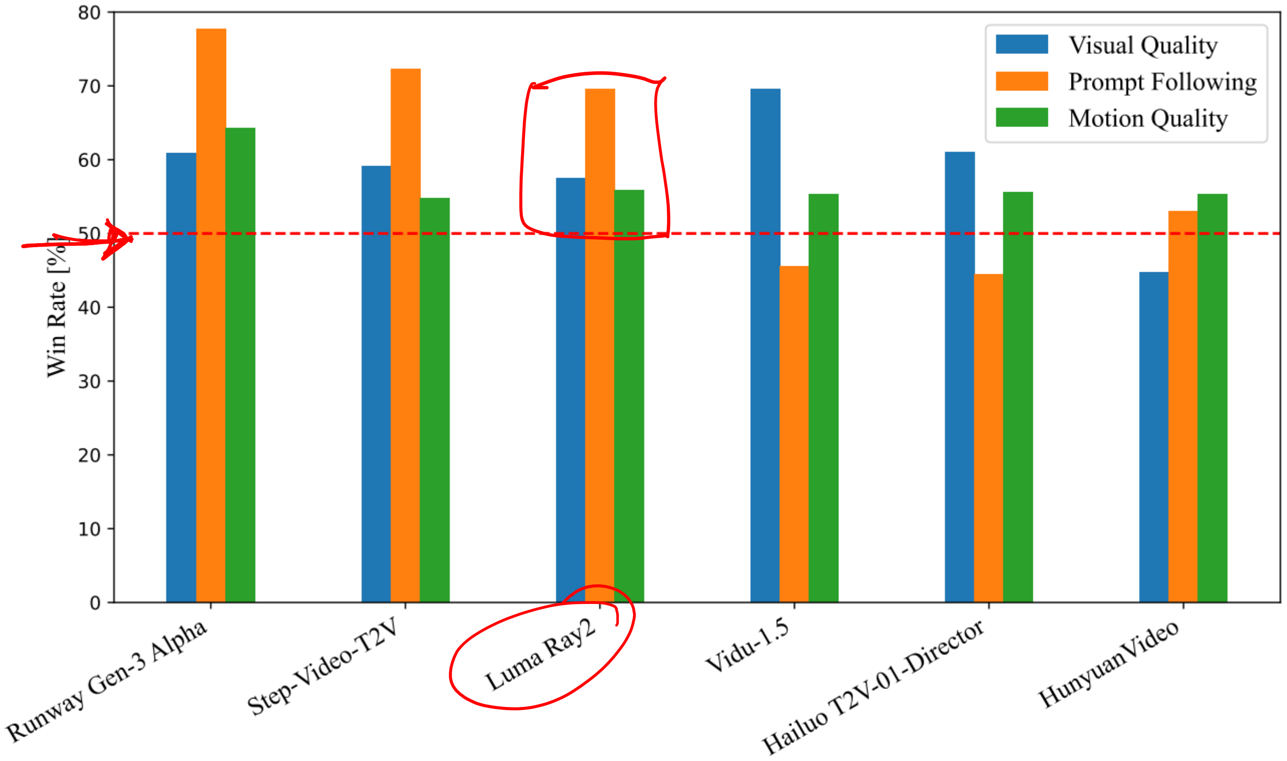


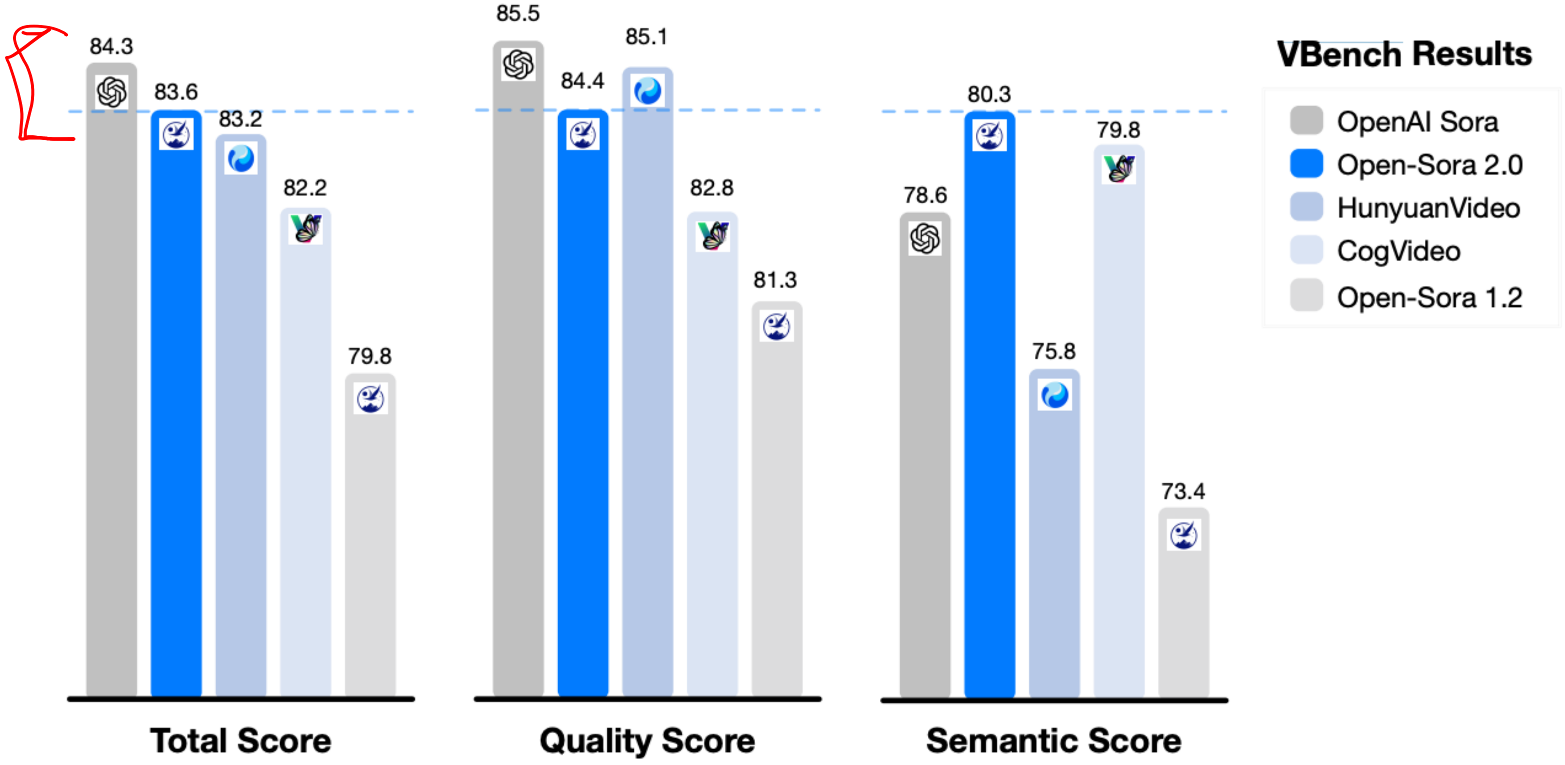
Figure 1: Human preference evaluation of Open-Sora 2.0 against other leading video generation models. Win rate represents the percentage of comparisons where our model was preferred over the competing model. The evaluation is conducted on 100 prompts carefully designed to cover three key aspects: 1) visual quality, 2) prompt adherence, and 3) motion quality. Results show that our model performs favorably against other top-performing models in all three aspects.

https://arxiv.org/pdf/2503.09642v1

OpenSora

How do we evaluate?

- Human evaluation
- Benchmarks + Metrics



https://arxiv.org/pdf/2503.09642v1

VIDEO AND VLMS

Large World Model

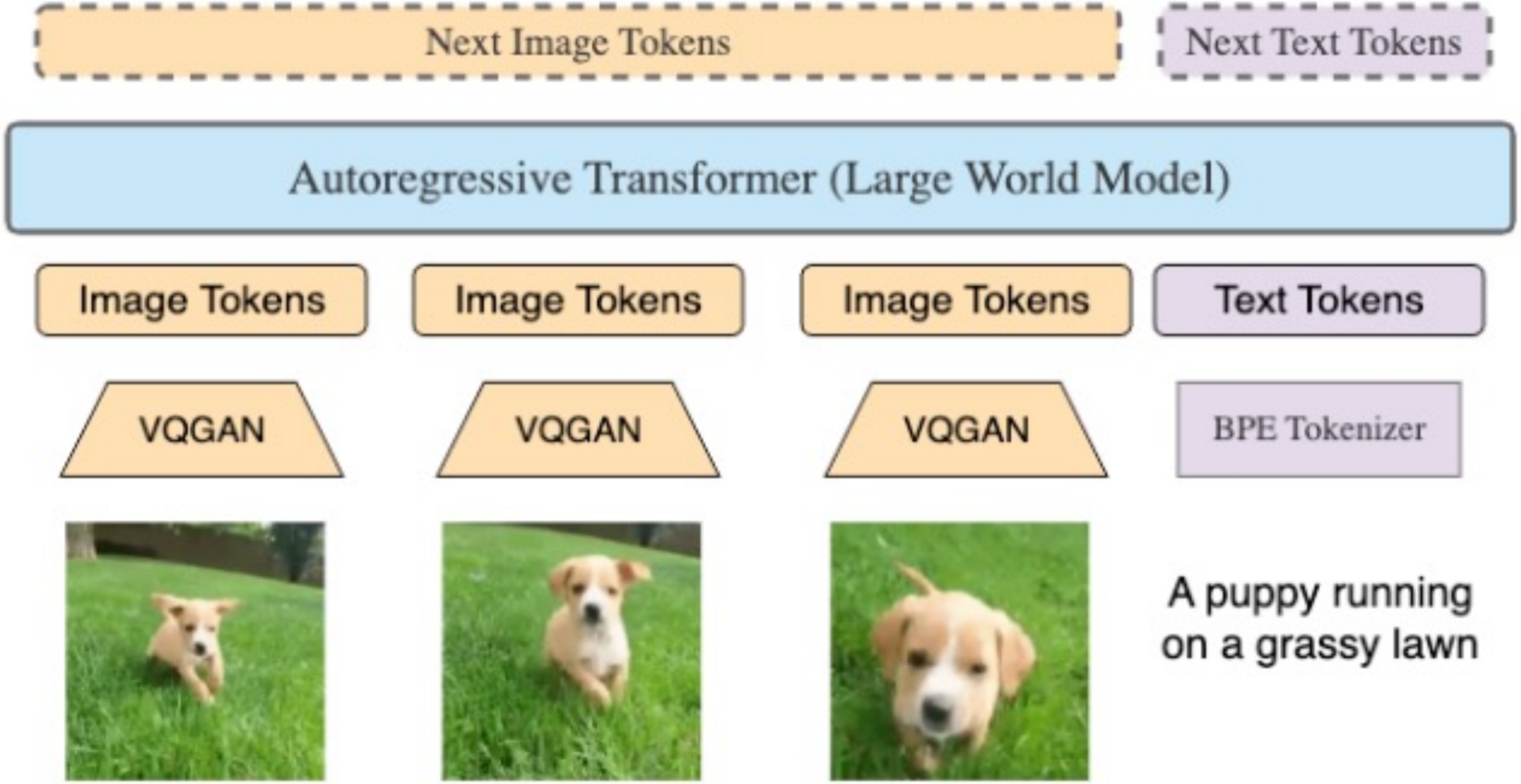
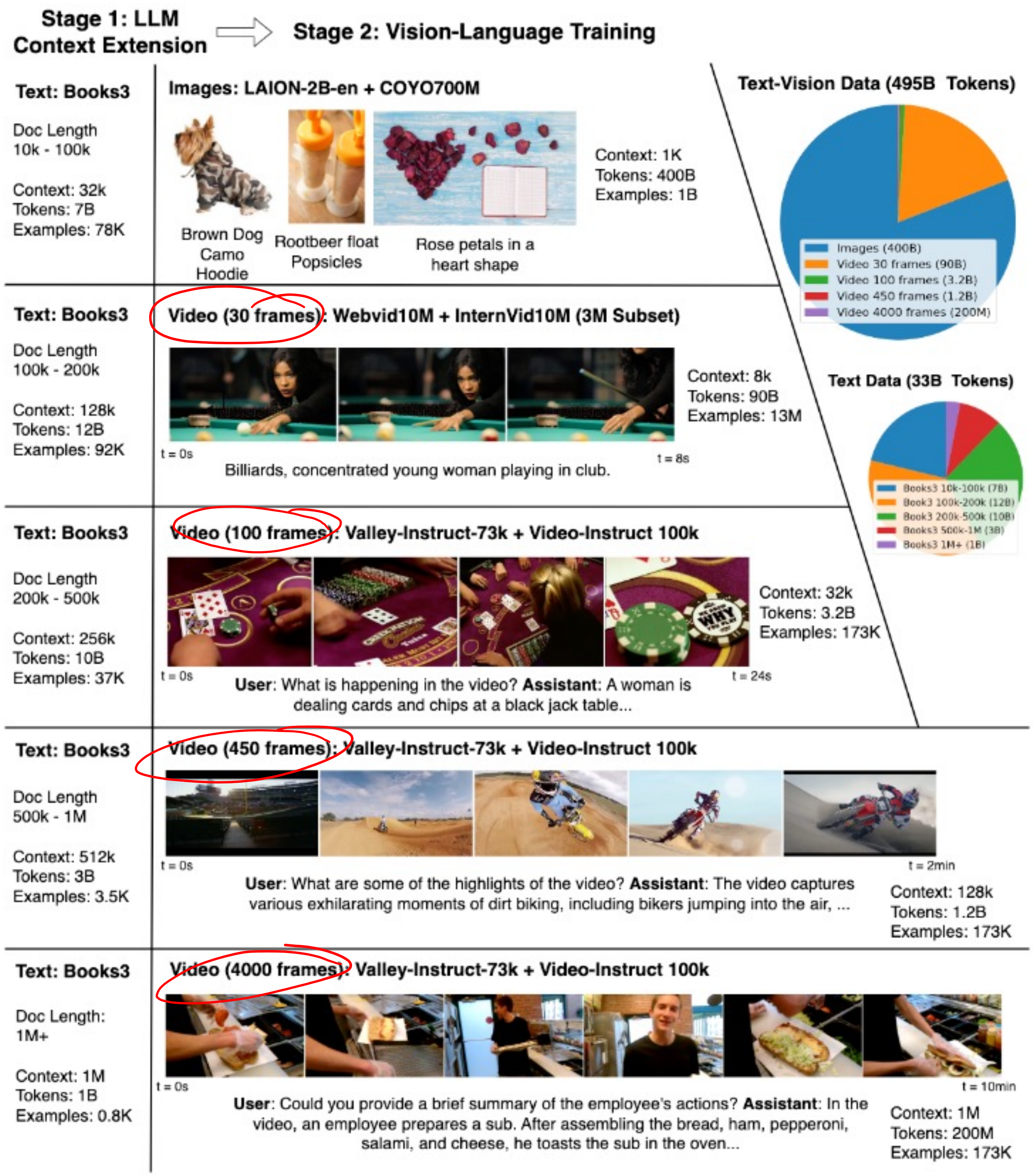


Figure from http://arxiv.org/abs/2402.08268

Large World Model



Billiards, concentrated young woman playing in club.

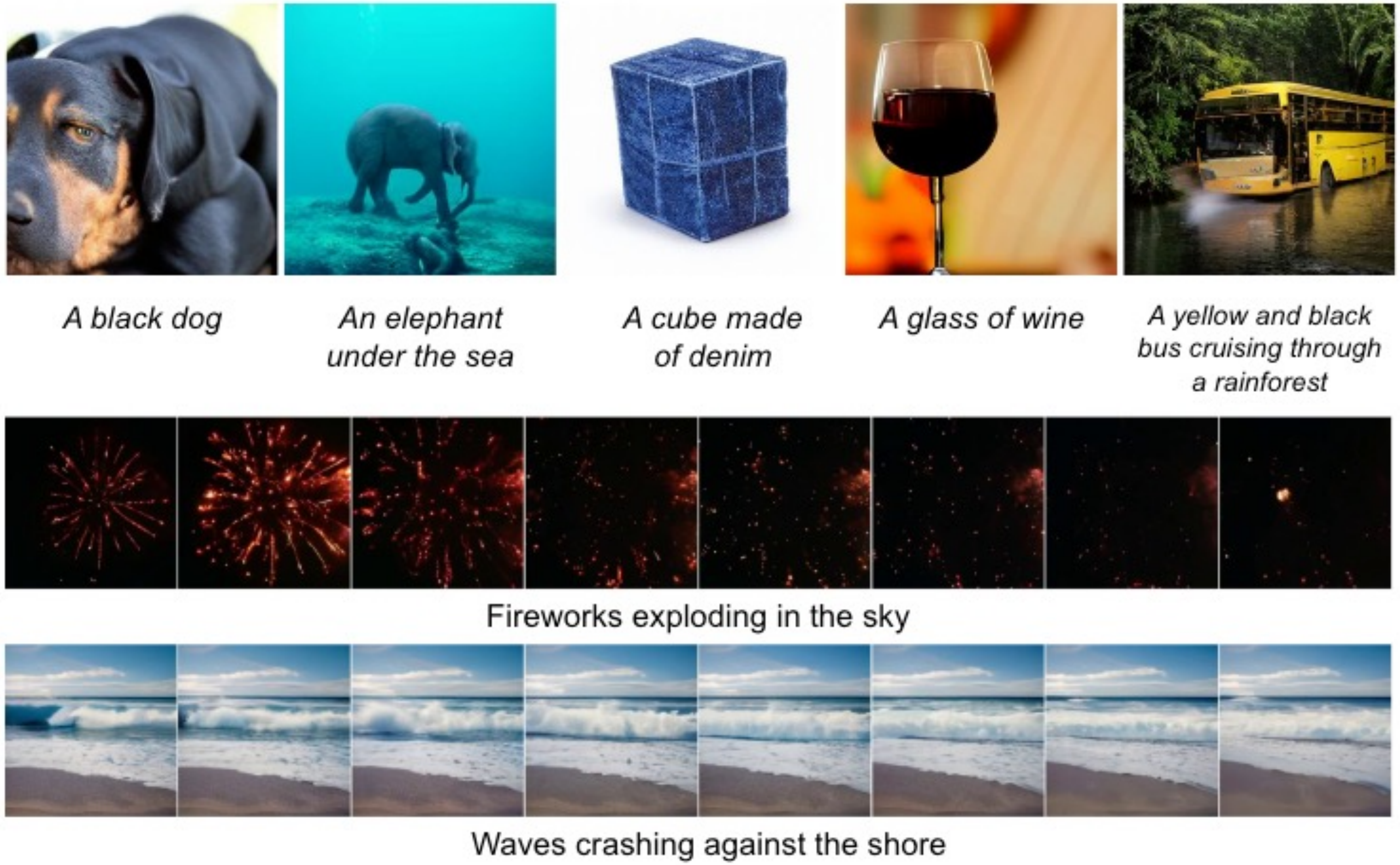
User: What is happening in the video? **Assistant**: A woman is dealing cards and chips at a black jack table...

User: What are some of the highlights of the video? **Assistant**: The video captures various exhilarating moments of dirt biking, including bikers jumping into the air, ...

User: Could you provide a brief summary of the employee's actions? **Assistant**: In the video, an employee prepares a sub. After assembling the bread, ham, pepperoni, salami, and cheese, he toasts the sub in the oven...

Figure from http://arxiv.org/abs/2402.08268

Large World Model



A black dog

An elephant under the sea

A cube made of denim

A glass of wine

A yellow and black bus cruising through a rainforest

Fireworks exploding in the sky

Waves crashing against the shore

Figure 8 LWM can generate images and videos given text input. Examples of image and video generations. More examples are shown in Appendix E and Appendix D.

Figure from http://arxiv.org/abs/2402.08268