SORA

Diffusion, applied to Video

Model Created by: OpenAI

Presented by:
John Tan Chong Min

Example Video

Prompt:
A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage.

She wears a black leather jacket, a long red dress, and black boots, and carries a black purse.

She wears sunglasses and red lipstick.

She walks confidently and casually.

The street is damp and reflective, creating a mirror effect of the colorful lights.

Many pedestrians walk about.

Does not fully understand physics

Prompt:

Step-printing scene of a person running, cinematic film shot in 35mm.

Spot the error?

SORA Overview

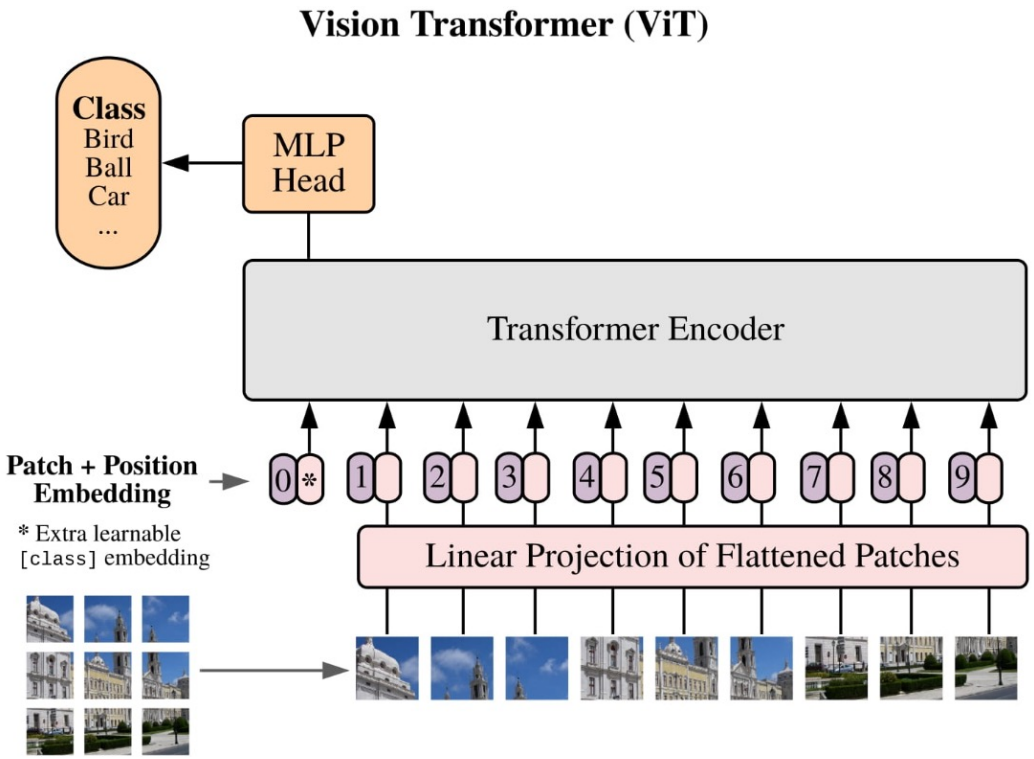
- **Diffusion model**: Starts off with image patches that looks like static noise and gradually transforms it by removing the noise over many steps
- Input can be entire video, or single image frame
- Works for different image resolution and aspect ratio (**huge breakthrough!**)

Transformer

- Traditional LLM: Next token prediction

- SORA: Next frame prediction as **patch prediction problem** conditioned on text of video description

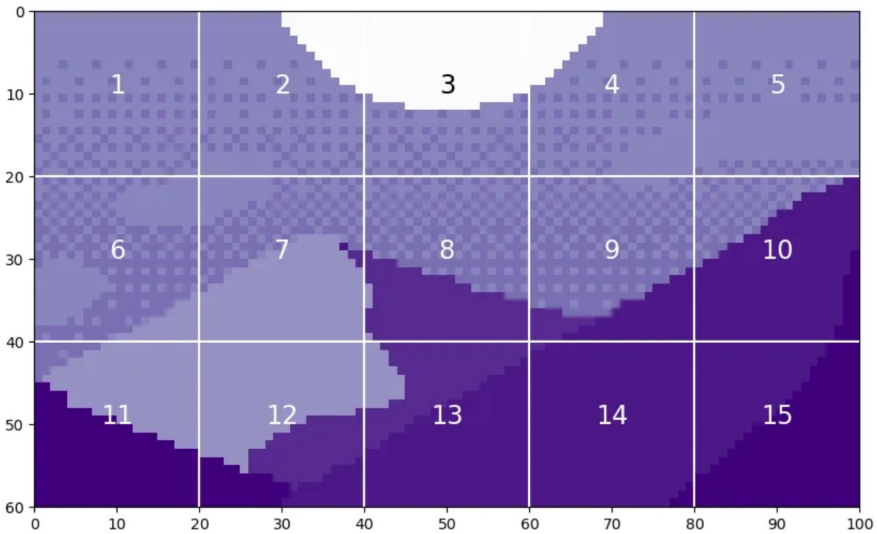
Vision Transformer (ViT) splits image into patches



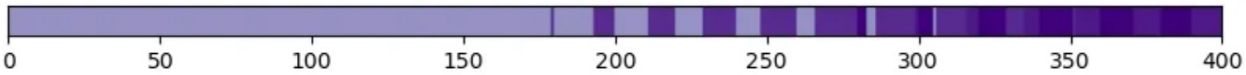
- Idea of splitting image into patches for tokenization is interesting
- But supervised objective meant it could not scale

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Dosovitskiy et. al. 2021.

Patch interpretation in ViT



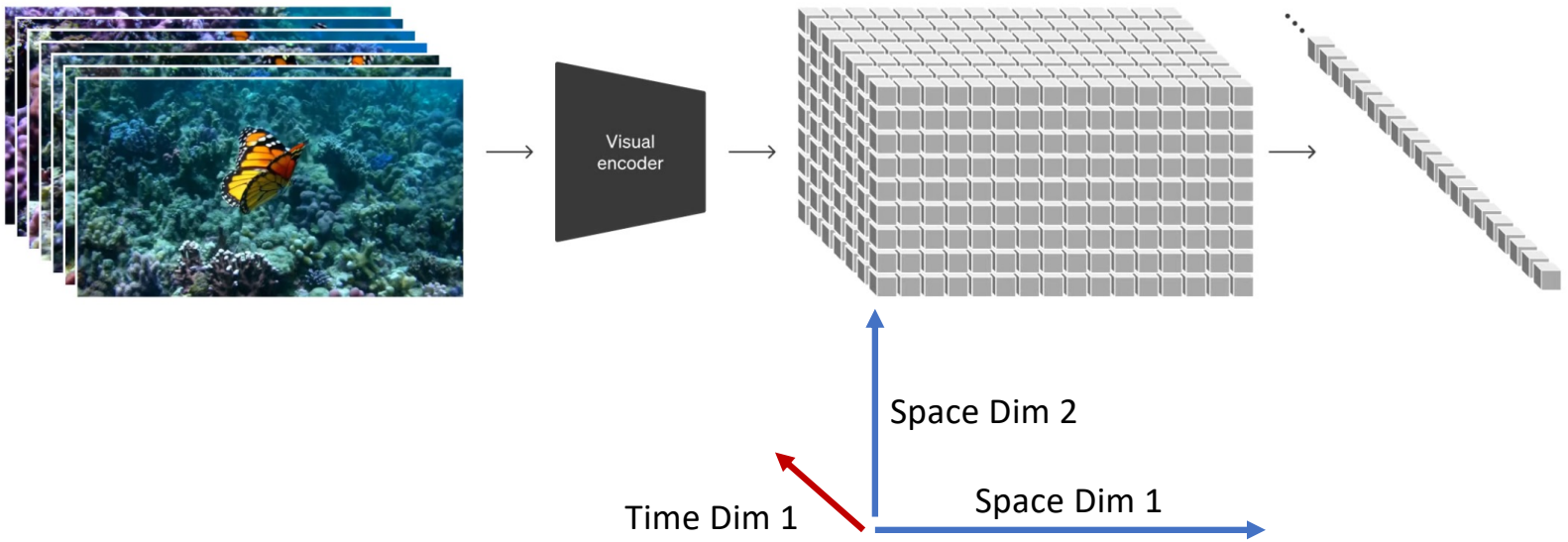
Patch 12 flattened



Final Patch can be linear
Projection of this flattened patch
(dim *n* - dimension of transformer embeddings)

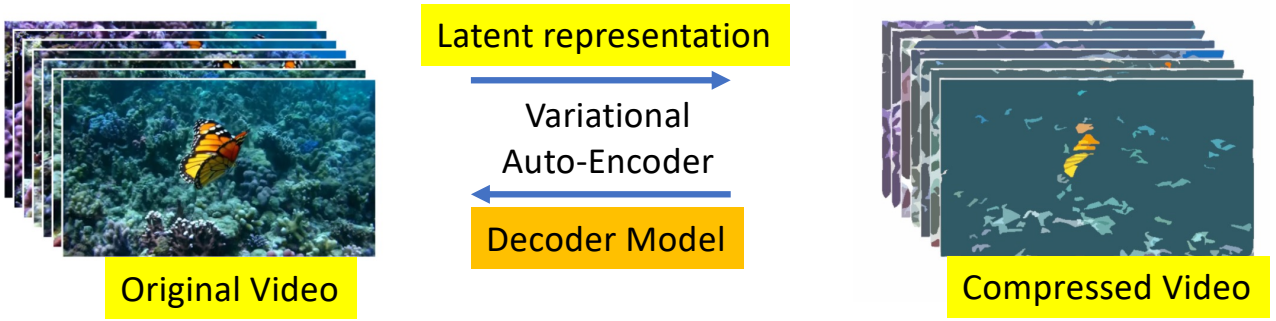
https://towardsdatascience.com/vision-transformers-explained-a9d07147e4c8

SORA splits multiple images into 3D patches



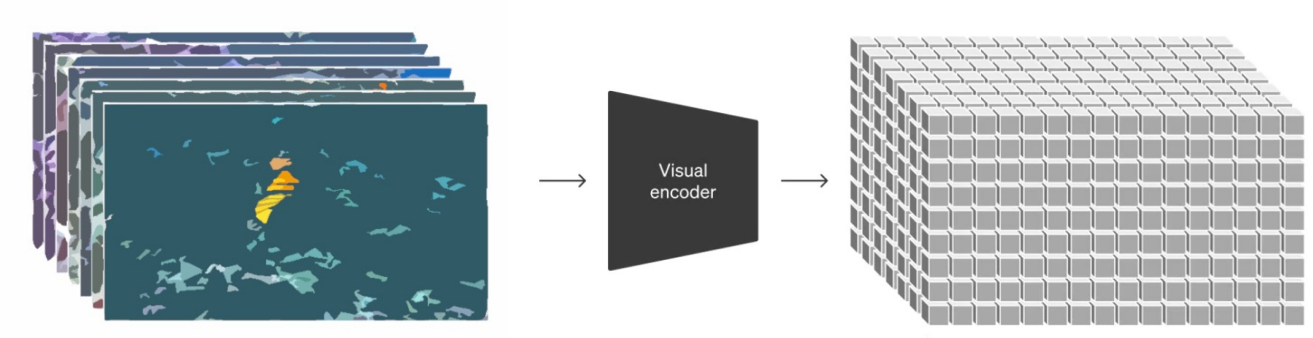
At a high level, we turn videos into patches by first compressing videos into a lower-dimensional latent space,[19] and subsequently decomposing the representation into spacetime patches.

Step 1: Video Compression



- We train a network that **reduces the dimensionality** of visual data
- This network takes raw video as input and outputs a latent representation that is compressed both temporally and spatially
- SORA is trained on and subsequently generates videos within this compressed latent space
- We also train a corresponding decoder model that maps generated latents back to pixel space

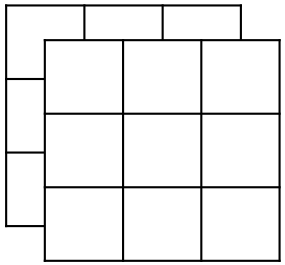
Step 2: Compressed Video -> Spacetime Patches



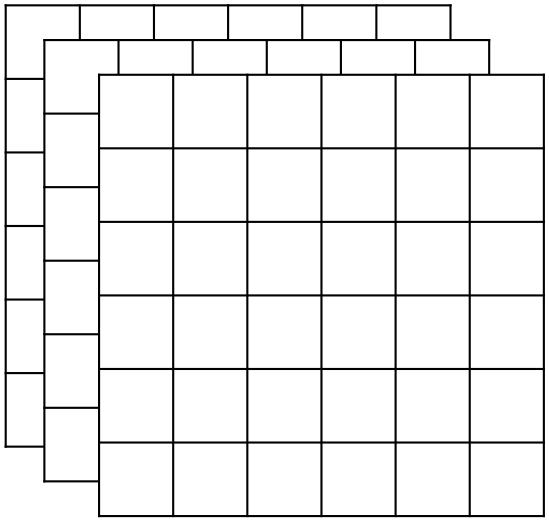
- Given a compressed input video, we extract a sequence of spacetime patches which act as transformer tokens
- This scheme works for images too since images are just videos with a single frame
- Our patch-based representation enables Sora to train on videos and images of variable resolutions, durations and aspect ratios
- At inference time, we can control the size of generated videos by **arranging randomly-initialized patches in an appropriately-sized grid**

Spacetime Patches works for any resolution and video length (up to 1 min)

- Diffusion Model can take in spacetime patches in array of multiple sizes
- Different sized arrays leads to different resolution of video, video duration

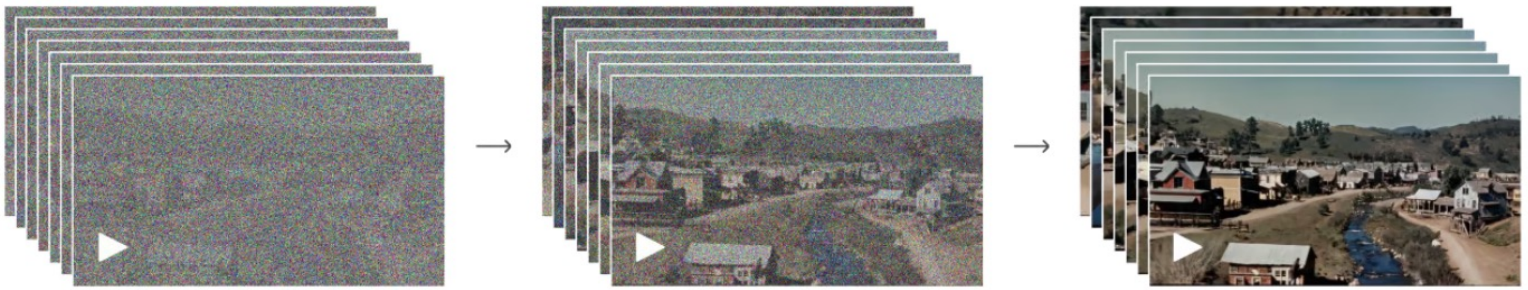


Low Res Video
Short Duration



High Res Video
Long Duration

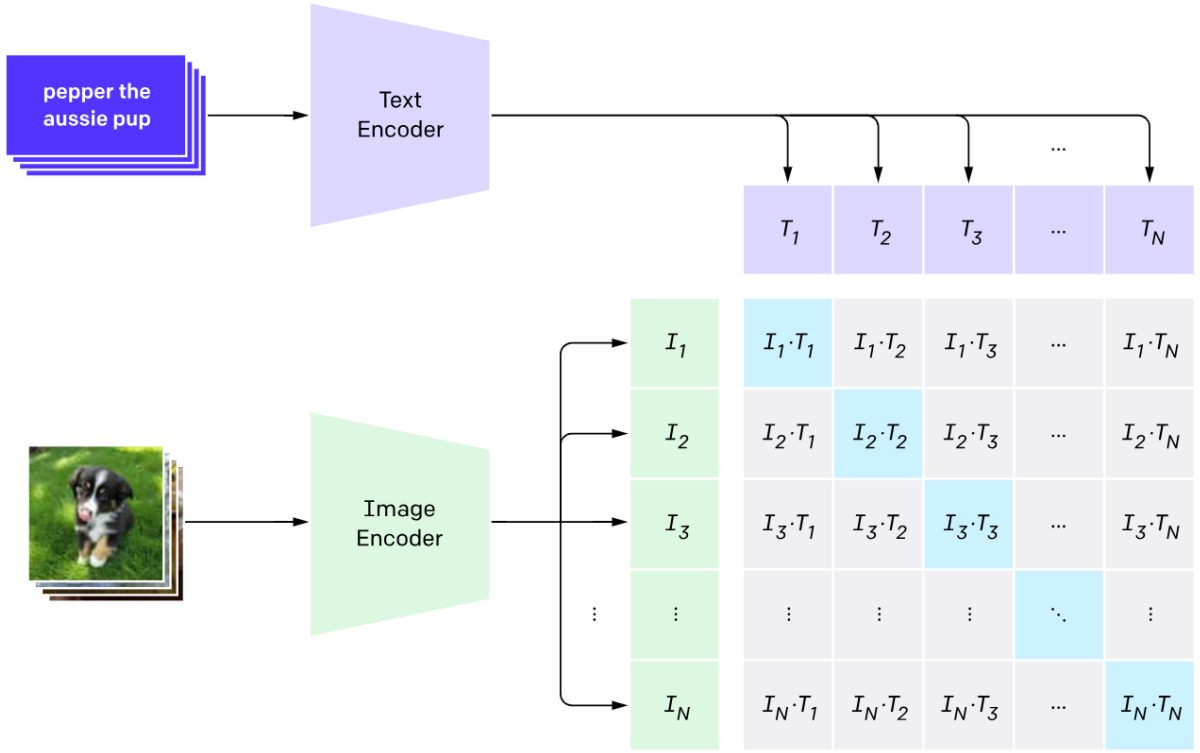
Diffusion Transformer



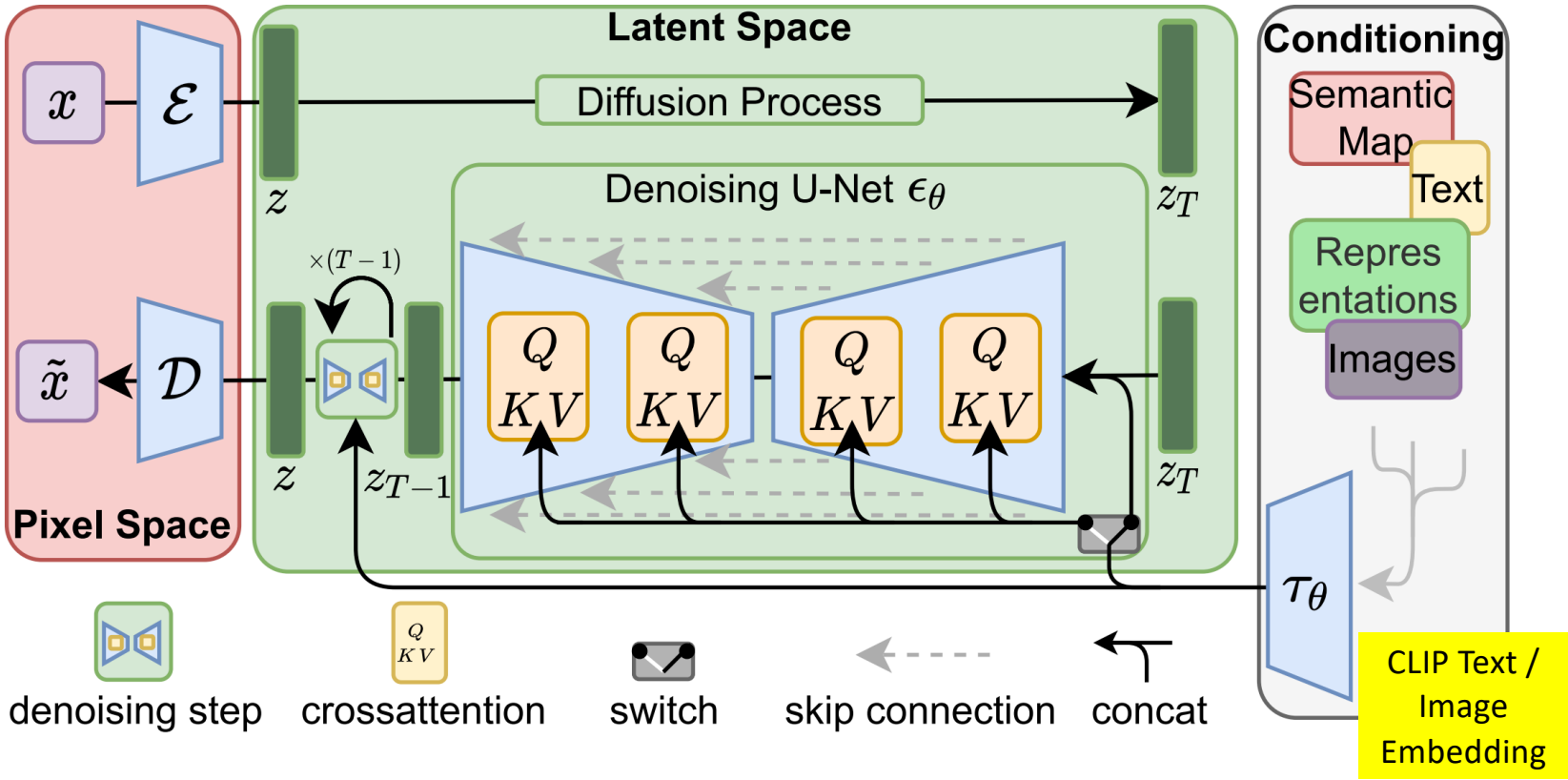
- Sora is a diffusion model
- Given input noisy patches (and conditioning information like text prompts), it's trained to predict the original "clean" patches.
- (My view) Can be trained in a self-supervised fashion by simply predicting next frame/previous frame in a video conditioned on text prompt

Recap: CLIP Embeddings to map Image to Text

- CLIP Embeddings learned by contrastive loss, **maps image and text to same representational space**



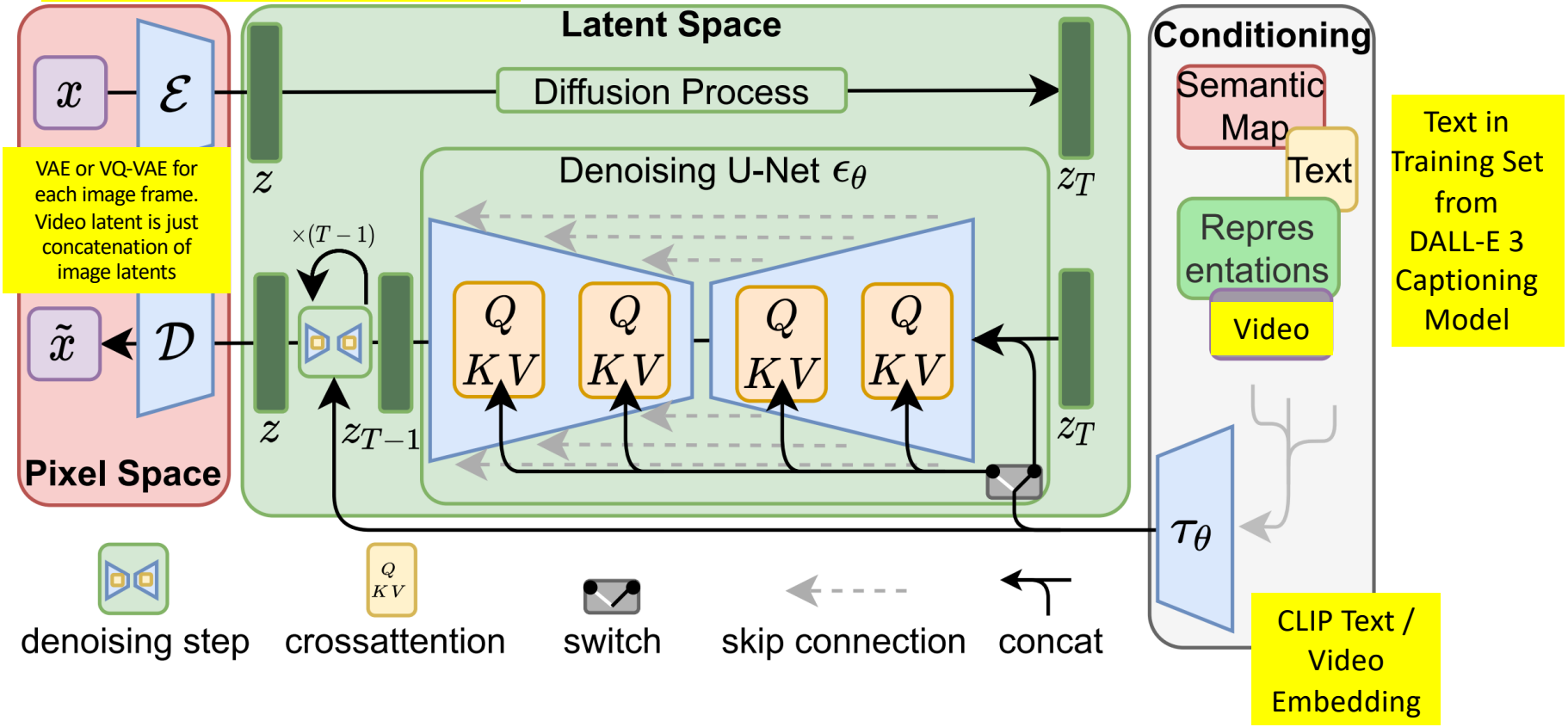
Stable Diffusion (Image)



High-Resolution Image Synthesis with Latent Diffusion Models. Robin et. Al. 2022.

Stable Diffusion (Video) – My Speculation

This z is video-level linear vector

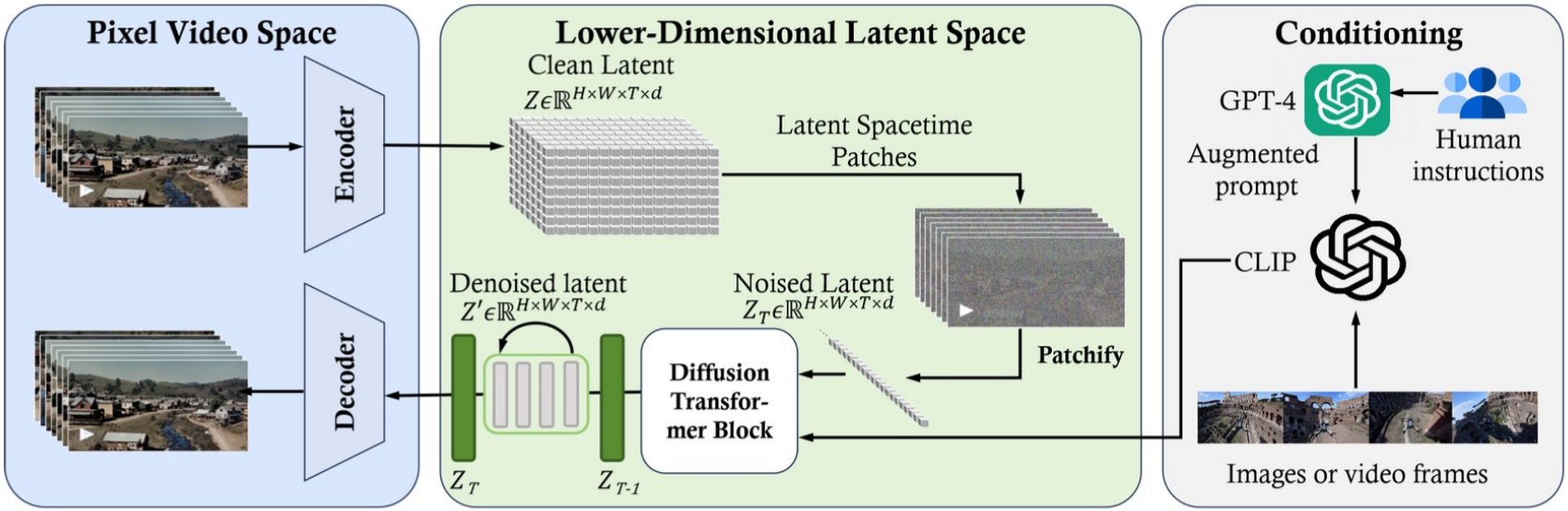


VAE or VQ-VAE for each image frame. Video latent is just concatenation of image latents

Text in Training Set from DALL-E 3 Captioning Model

CLIP Text / Video Embedding

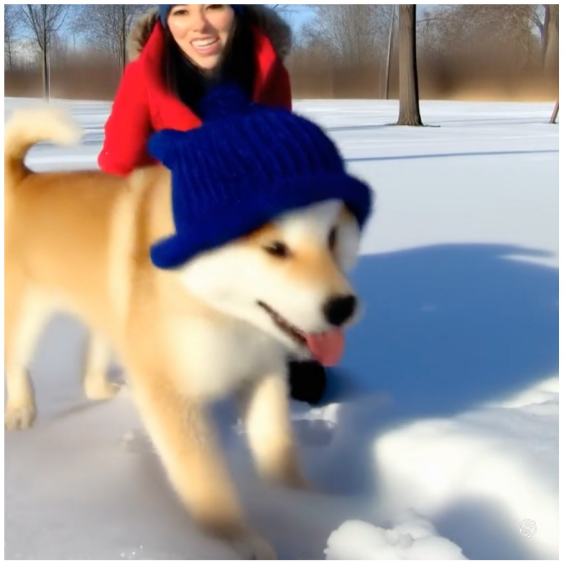
Putting it all together



Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models. Liu et. al. 2024.

Diffusion Generates Better Videos

Base compute



4x compute

32x compute

More Speculation

- Could something like **ControlNet** (Zhang et. al., 2023) be used to edit videos in certain positions, or in certain styles, much like how it is done for images?

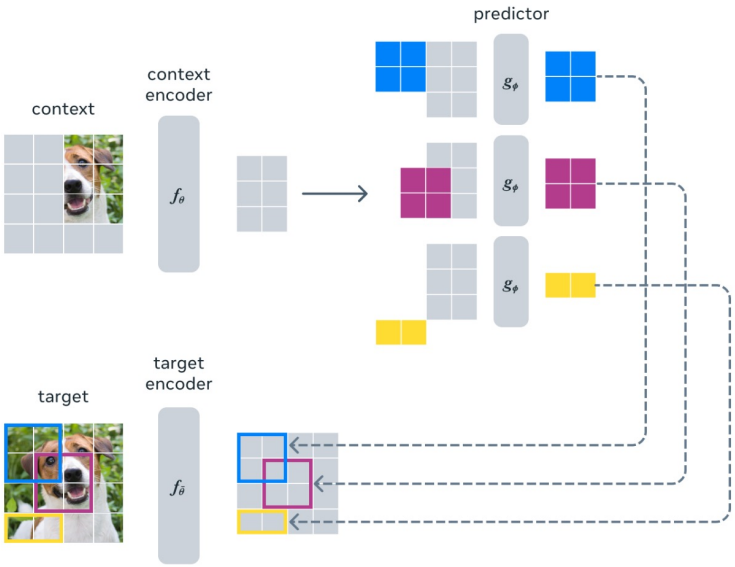
Questions to Ponder

- How can we get compressed latent space from video effectively?
- How can CLIP for text encompass so much nuance about the video description?
- Would SORA be a good simulator?
- How can we imbue physics know-how into SORA?
- So far, video generation is taking a low-dimensional text and mapping it into video. How can we do the reverse and **use video to get the corresponding text**. There are multiple possible abstractions for the text, how do we get these multiple abstraction spaces?

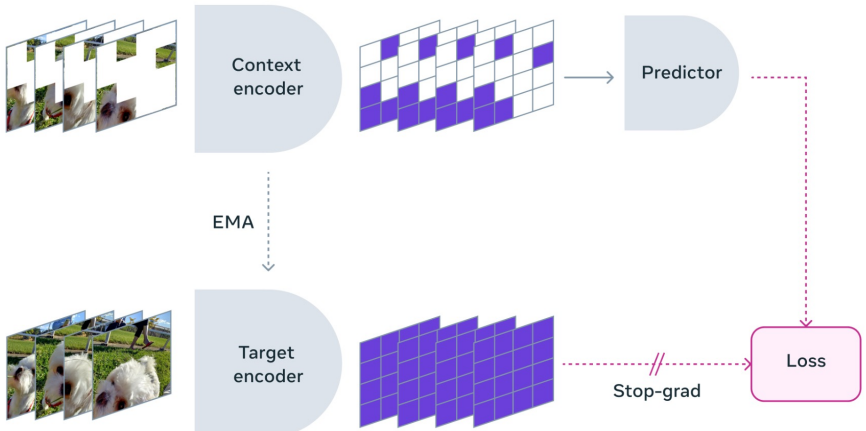
Prior work on video latent spaces

I-JEPA / V-JEPA (META)

- Predicting within same image frame is not as powerful as next-frame prediction
- Lack of text input conditioning for generation of video from text



I - JEPA



V - JEPA

I-JEPA / V-JEPA. Meta. 2023/2024.

Make-a-Video (META)

- Primarily trained by Text-to-Image pairs and expanding it out in temporal space
- May lack overall video context as it does not use Text-to-Video training

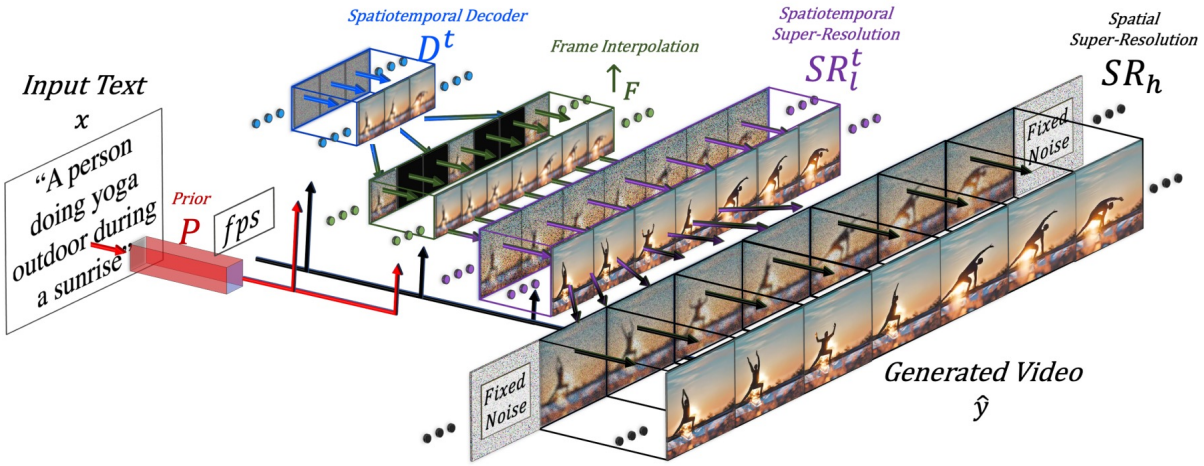


Figure 2: **Make-A-Video high-level architecture.** Given input text x translated by the prior P into an image embedding, and a desired frame rate fps, the decoder D^t generates 16 64×64 frames, which are then interpolated to a higher frame rate by $\uparrow_F$, and increased in resolution to 256×256 by SR_l^t and 768×768 by SR_h, resulting in a high-spatiotemporal-resolution generated video $\hat{y}$.

Make-A-Video: Text-to-Video Generation without Text-Video Data. Meta. 2022.

Taking insights from DALL-E 2 (Can there be a Video-Text CLIP?)

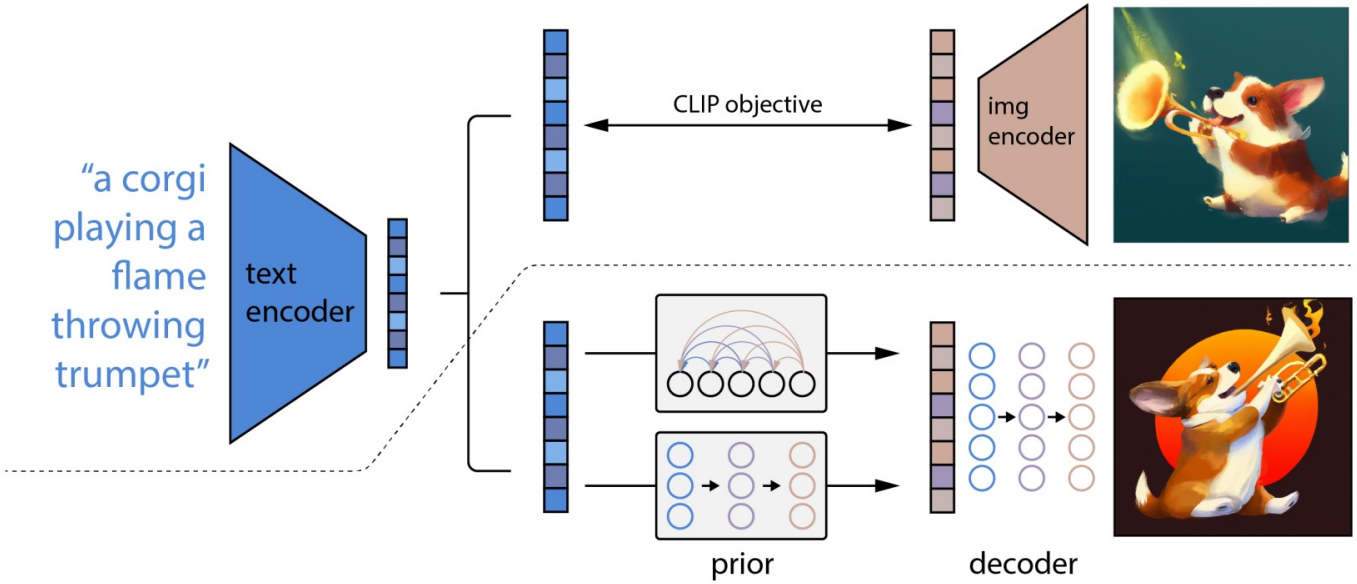


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

DALL E-2 preserves the original text for diffusion prior

For the diffusion prior, we train a decoder-only Transformer with a causal attention mask on a sequence consisting of, in order: the encoded text, the CLIP text embedding, an embedding for the diffusion timestep, the noised CLIP image embedding, and a final embedding whose output from the Transformer is used to predict the unnoised CLIP image embedding. We choose not to condition the diffusion prior on $z_i \cdot z_t$ like in the AR prior; instead, we improve quality during sampling time by generating two samples of z_i and selecting the one with a higher dot product with z_t. Instead of using the ϵ-prediction formulation from Ho et al. [25], we find it better to train our model to predict the unnoised z_i directly, and use a mean-squared error loss on this prediction:

$$L_{\text{prior}} = \mathbb{E}_{t \sim [1,T], z_i^{(t)} \sim q_t} \left[\| f_\theta(z_i^{(t)}, t, y) - z_i \|^2 \right]$$

Hierarchical Text-Conditional Image Generation with CLIP Latents (DALL-E 2). OpenAI. 2022.