CogVideoX

Text-to-Video Diffusion Models with An Expert Transformer

2024.12.31 | 카피바라팀 | 배누리, 김호정, 전사영, 박현아

CONTENTS

01

Introduction

DiT

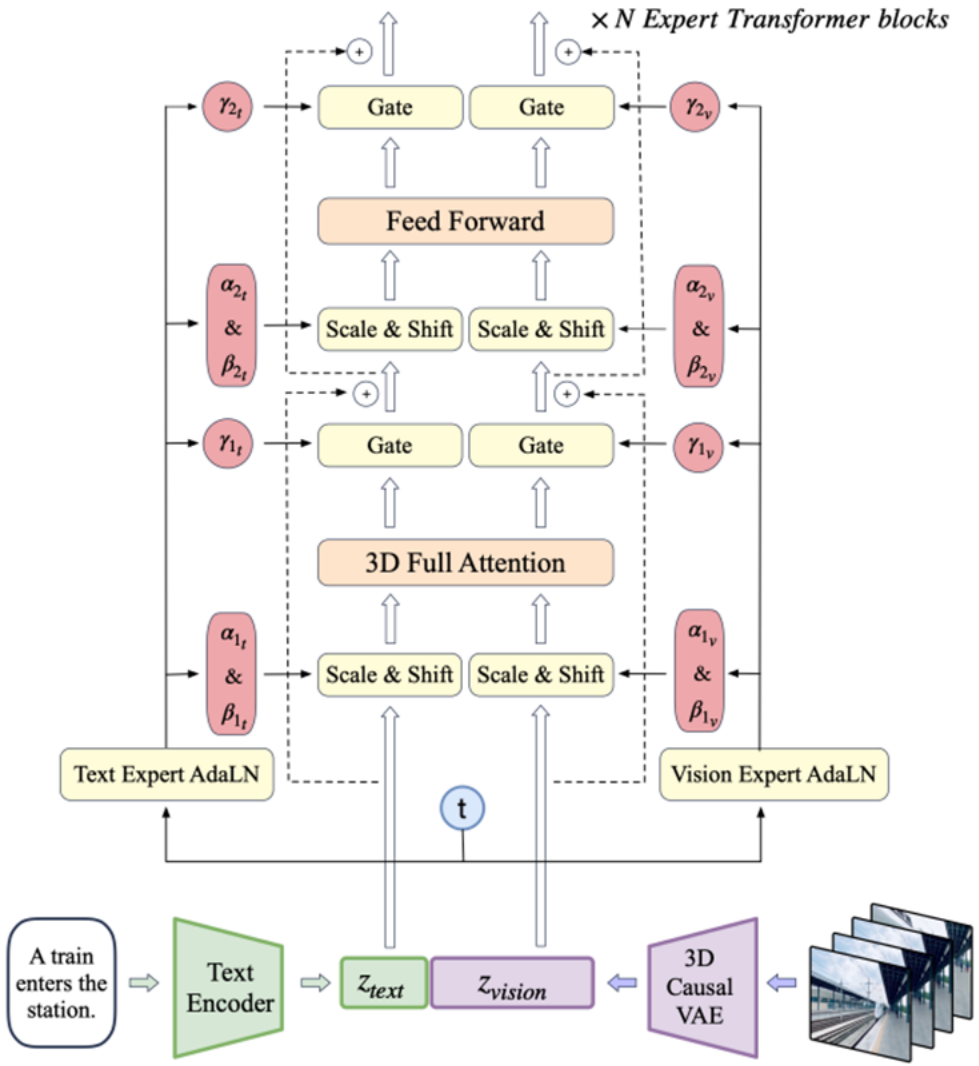
- Diffusion Transformers (DiT)를 사용함으로써 text-to-video 생성은 획기적인 수준에 도달
- 장기적으로 일관된 동영상을 생성하는 방법은 기술적으로 불분명
- 다음과 같은 여러 과제들이 지금까지 대체로 해결되지 않음
  - 효율적인 동영상 데이터 모델링
  - 효과적인 텍스트-동영상 정렬
  - 모델 학습을 위한 고품질 텍스트-동영상 쌍 구성

CogVideoX

3D VAE, expert Transformer, 동영상 데이터 필터링 및 captioning 파이프라인을 개발하여 해결

- 효율적인 동영상 데이터 모델링
  → 3D causal VAE
- 효과적인 텍스트-동영상 정렬
  → Expert Adaptive Layernorm을 갖춘 expert Transformer를 사용
- 모델 학습을 위한 고품질 텍스트-동영상 쌍 구성
  → 동영상 콘텐츠를 정확하게 설명할 수 있는 동영상 captioning 파이프라인을 개발

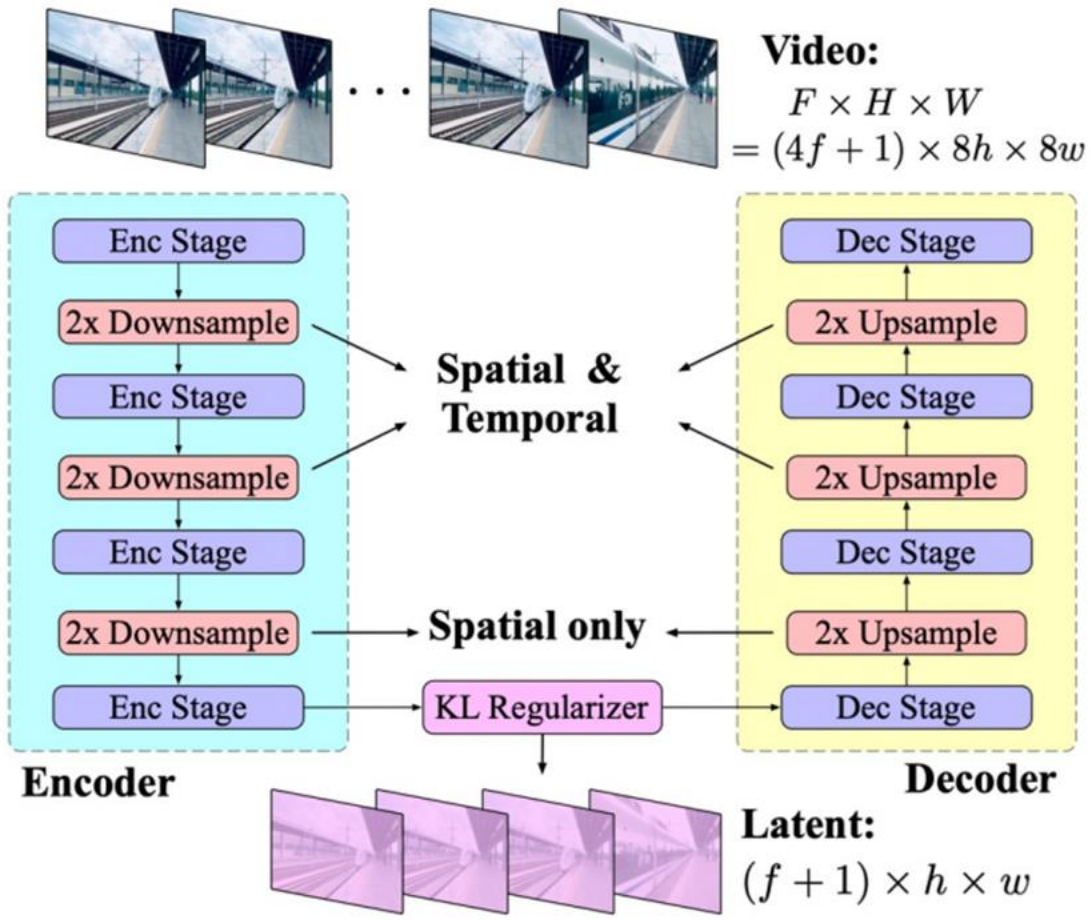
Architecture



1. 동영상과 텍스트 입력이 주어지면, 3D causal VAE로 동영상을 latent space로 압축
2. Latent 데이터는 Patchify되어 긴 시퀀스 z_{vision}로 펼쳐짐
3. T5를 사용하여 텍스트 입력을 텍스트 임베딩 z_{text}로 인코딩
4. z_{text}와 z_{vision}은 시퀀스 차원을 따라 concat함
5. Concat된 임베딩은 expert transformer block의 스택에 입력
6. 모델 출력은 unpatchify되어 원래 latent 모양으로 복원
7. 3D causal VAE 디코더를 사용하여 디코딩되어 동영상을 재구성

Architecture

3D Causal VAE

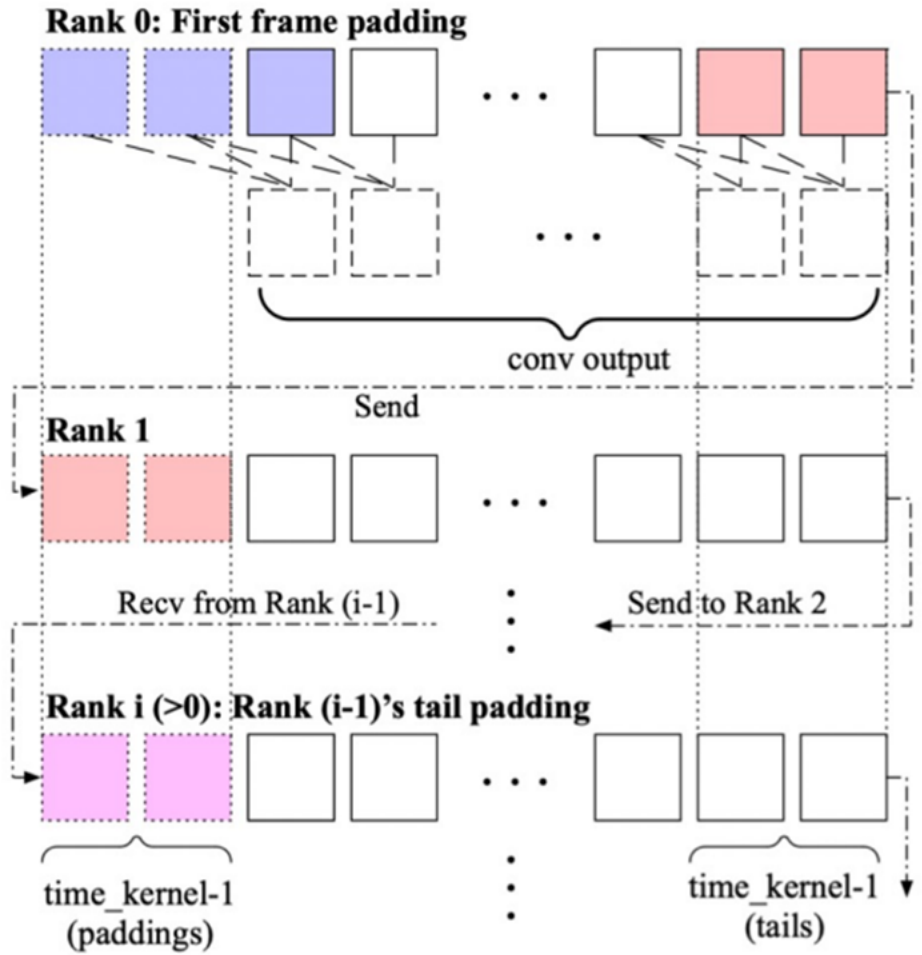


- 인코더, 디코더, latent space regularizer로 구성
- latent space는 KL regularizer에 의해 제한
- 인코더와 디코더는 대칭적으로 배열된 4개의 stage로 구성되어 있으며, 각각 ResNet block들의 스택으로 2배 다운샘플링 및 업샘플링을 수행
- 인코더의 처음 두 다운샘플링과 디코더의 마지막 두 업샘플링은 시공간 차원에 모두 적용되는 반면, 인코더의 마지막 다운샘플링과 디코더의 첫 번째 업샘플링은 공간 차원에만 적용

  →3D VAE는 시간 차원에서 4배, 공간 차원에서 8×8 압축을 달성

Architecture

3D Causal VAE



Temporally Causal Convolution

- Causal Convolution:
  - 과거 프레임 정보만을 사용하여 현재 프레임을 처리
  - 미래 정보가 현재 또는 과거 예측에 영향을 미치지 않도록 보장
- Temporal Padding:
  - Convolution의 경계에서 데이터 손실을 방지하기 위해 프레임 시작 부분에 패딩을 추가
  - 첫 번째 프레임은 충분한 패딩을 받고, 이후는 순차적으로 이어지는 구조
- 각 Rank는 단순히 길이가 k−1인 세그먼트를 다음 Rank로 전달
  - k: Temporal kernel size (시간축에서 Convolution 필터 크기)
- Convolution은 이전 데이터와 새롭게 추가된 데이터가 겹치는 방식으로 처리

Architecture

Expert Transformer

- Patchify

1. Latent Space 의 구조

T×H×W×C 모양의 동영상 latent space를 인코딩

- T: 프레임 수 (시간적 차원)
- H: 각 프레임의 높이(Height)
- W: 각 프레임의 너비(Width)
- C: 채널 수

2. 패치화

Latent Space를 작은 패치(patch) 단위로 변환하여 시퀀스로 변환

z_{vision}의 길이= $\frac{T}{q} \times \frac{H}{p} \times \frac{W}{p}$

- q: 시간 차원을 패치로 분할하는 크기
- p: 공간 차원(높이와 너비)을 패치로 분할하는 크기

패치 단위로 나눠진 데이터는 1차원 시퀀스로 변환되어

Transformer에서 처리

Expert Transformer

- **3D-RoPE**
  - RoPE (Rotary Position Embedding) : LLM(대규모 언어 모델)에서 토큰 간 상대적 위치 정보를 효율적으로 인코딩하기 위해 사용되는 기법
  - 동영상 데이터에 RoPE를 사용하기 위해 3D-RoPE로 확장
  - 동영상 텐서의 각 latent는 3D 좌표 (x, y, t) 로 표현할 수 있음
  - 좌표의 각 차원에 1D-RoPE를 독립적으로 적용
    - x: 전체 hidden states 채널의 $\frac{3}{8}$ 사용
    - y: 전체 hidden states 채널의 $\frac{3}{8}$ 사용
    - t: 전체 hidden states 채널의 $\frac{2}{8}$ 사용 (시간적 정보는 상대적으로 더 작은 비중)
  - 결과적으로 생성된 각 RoPE 인코딩은 채널 차원에서 concat되어 최종적인 3D-RoPE 인코딩을 만듦.
  - 공간 및 시간적 정보를 분리하여 각각 독립적으로 처리

Architecture

Expert Transformer

- **Expert Adaptive Layernorm**

- 텍스트와 비디오 데이터는 서로 다른 형태의 정보를 담고 있음
  - 텍스트 데이터 : 문장 구조와 단어 간의 관계를 포함
  - 비디오 데이터 : 시간적(Temporal) 및 공간적(Spatial) 정보를 포함
- 이러한 특성 차이로 인해, 두 데이터를 Transformer에서 결합하여 학습하는 데 어려움이 발생
  - 문제: 두 모달리티의 스케일와 공간적 특징 차이

Expert Adaptive Layernorm (AdaLN)

- 모달리티별 독립적 처리
  - Vision Expert AdaLN: 비디오 데이터의 hidden states 조정
  - Text Expert AdaLN: 텍스트 데이터의 hidden states 조정
- 학습 중 데이터의 타임스텝(시간적 정보)에 따라 조정

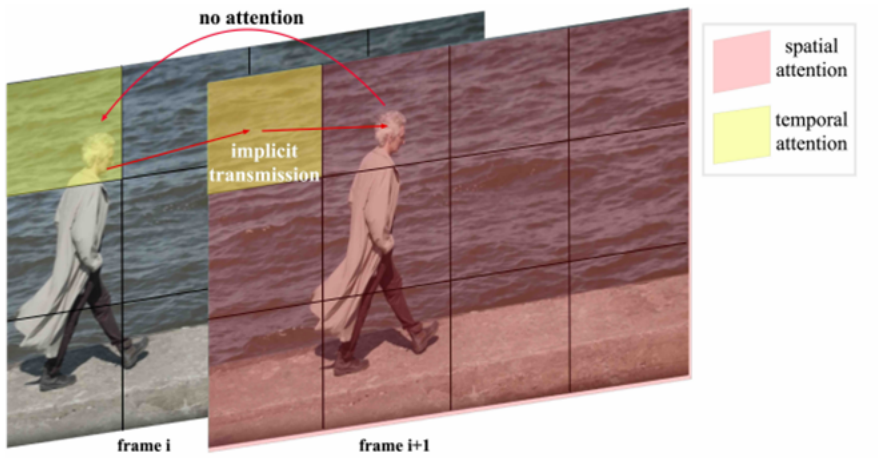
이점

- 특징 공간 정렬 개선 : 텍스트와 비디오의 특징 공간 차이를 줄이고 통합.
- 효율적 처리 : 추가적인 파라미터 증가 없이 높은 성능.
- 확장성 : 다양한 모달리티 데이터에 적용 가능.

Architecture

Expert Transformer

- 3D Full Attention



기존 연구에서는 공간 정보(Spatial Attention)와 시간 정보(Temporal Attention)를 별도로 처리

- 큰 움직임이 있는 객체(예: 프레임 i+1 의 사람 머리)가 직접 연결되지 못함

  → 학습 복잡도 증가: 모델이 정보를 명확히 연결하지 못해 학습이 어려움.

  → 비일관성 : 큰 움직임이 있는 객체의 시각적 정보가 일관되지 않게 생성될 가능성

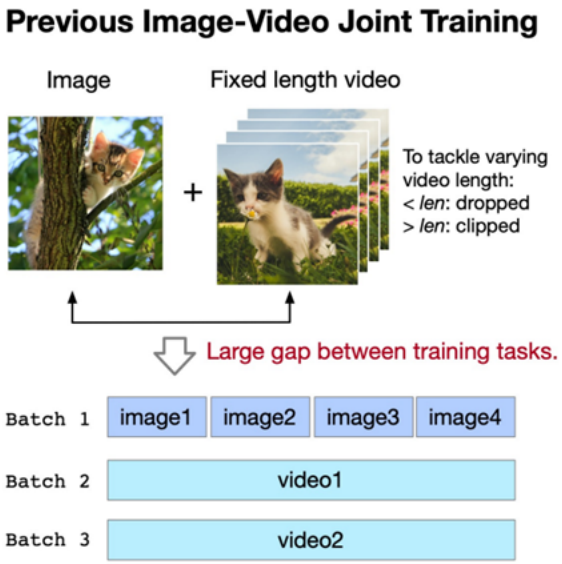
⬇

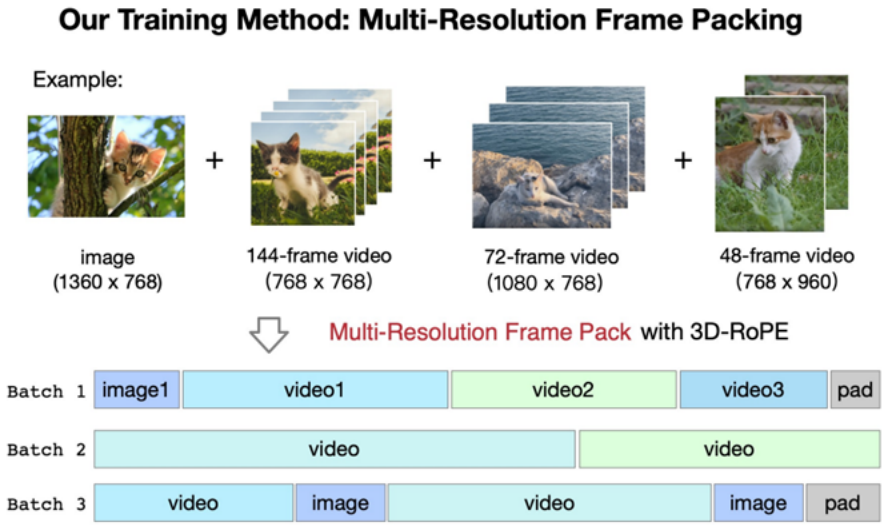
3D Text-Video Hybrid Attention 제안

- Spatial과 Temporal Attention을 통합하여 시공간적 상관관계를 명확히 학습
- 성능 향상 및 병렬 처리에 용이.

Training CogVideoX

Multi-Resolution Frame Pack





- 이미지와 고정된 프레임 수를 가진 동영상을 함께 학습.
- 문제점
  - 동영상은 여러 프레임으로 구성되어 있으나, 이미지는 한 프레임만 포함
    → 이미지와 동영상 사이에 학습 격차가 발생.
  - 다양한 길이를 가진 동영상을 충분히 학습하지 못함. 짧은 동영상은 늘리고, 긴 동영상은 잘라내야 하는 비효율적인 과정 발생.

- 혼합 길이 학습 방식을 도입하여 서로 다른 길이의 동영상을 함께 학습.
- 동영상 데이터를 다양한 프레임 길이로 변환하여 학습 효율을 극대화.
- Multi-Resolution Frame Pack 방식에 3D-RoPE를 적용하여 시간과 공간 정보를 효과적으로 학습.

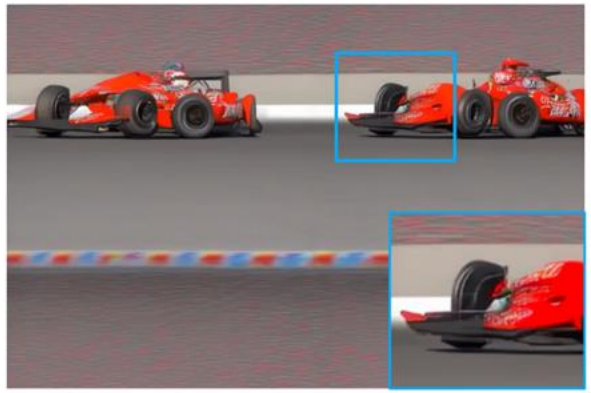
Training CogVideoX

Progressive Training

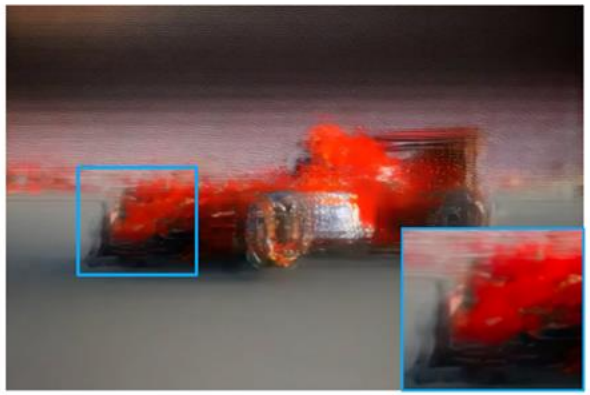
인터넷 동영상에는 일반적으로 상당한 양의 저해상도 동영상이 포함 , 고해상도 비디오에서 직접 훈련하는 것은 비용이 매우 큼.

→ 단계적 해상도 훈련 진행 (저해상도 학습, 고해상도 학습, 고품질 동영상 fine-tuning의 세 단계)

1. 256px 해상도 비디오로 먼저 훈련하여 의미적 정보(semantic knowledge)와 저주파 정보(low-frequency knowledge)를 학습.
2. 해상도를 점진적으로 증가시키며 훈련 : 256px → 512px → 768px.
3. 비디오의 화면 비율(aspect ratio)을 유지하며 짧은 쪽을 각 해상도로 크기 조정.
4. 고품질 데이터로 미세 조정(Fine-Tuning) – 자막과 워터마크 제거, 시각적 품질 향상
5. 이 과정을 기반으로 이미지에서 비디오를 생성하는 모델도 추가로 훈련



RoPE Extrapolation



RoPE Interpolation

저해상도 위치 인코딩을 고해상도로 조정할 때, 저자들은 interpolation과 extrapolation이라는 두 가지 다른 방법을 고려

- Interpolation : 글로벌 정보를 더 효과적으로 보존
- Extrapolation : 로컬한 디테일을 더 잘 유지

RoPE가 상대적 위치 인코딩이라는 점을 감안할 때, 저자들은 픽셀 간의 상대적 위치를 유지하기 위해 extrapolation을 선택

Training CogVideoX

DATA

- **Video Filtering**

부정적인 레이블을 정의

- 편집 영상 : 재편집이나 특수효과 등 명백히 인위적인 처리를 거친 영상.
- 모션 연결 부족 : 이미지 전환이 일어나고 모션 연결이 부족한 영상, 일반적으로 이미지를 인위적으로 이어 붙이거나 편집한 영상.
- 저품질 : 영상이 선명하지 않거나 카메라 흔들림이 심한 엉성한 촬영 영상.
- 강의 영상 : 교육 콘텐츠, 강의, 라이브 스트리밍 토론 등 적은 동작으로 지속적으로 대화하는 사람에 주로 초점을 맞춘 영상.
- 텍스트가 지배적인 영상 : 눈에 보이는 텍스트가 상당히 많거나 주로 텍스트 콘텐츠에 초점을 맞춘 영상.
- 노이즈가 많은 영상 : 휴대전화나 컴퓨터 화면에서 녹화된 노이즈가 많은 영상.

20,000개의 동영상 데이터 샘플을 샘플링하고 각각에 부정적인 태그가 있는지에 대한 레이블을 붙임

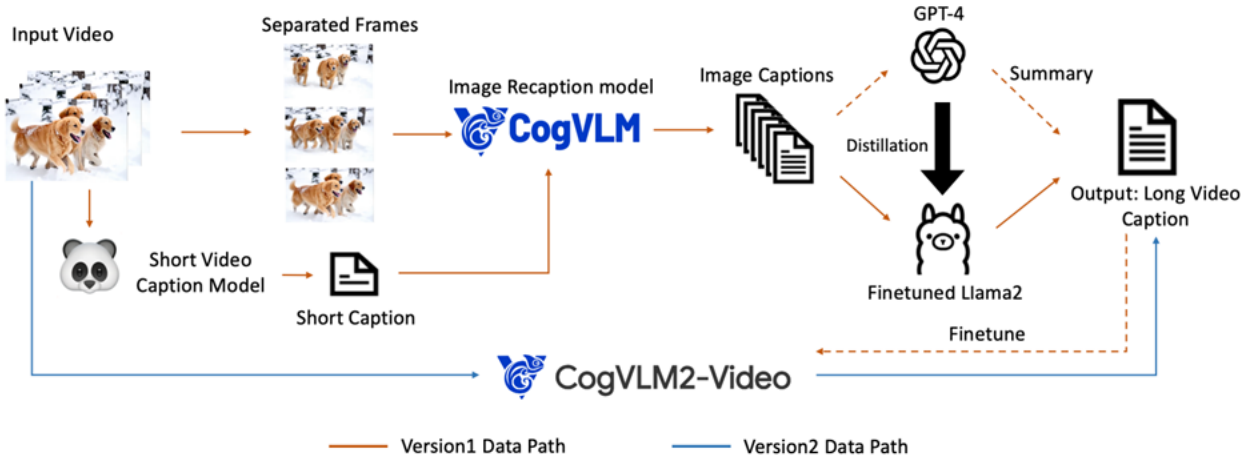
이러한 주석들을 사용하여 Video-LLaMA에 기반한 여러 필터를 학습시켜 저품질 동영상 데이터를 걸러냄

Training CogVideoX

DATA

- **Video Captioning**

대부분의 동영상 데이터에는 텍스트 설명이 제공되지 않으므로 동영상 데이터를 텍스트 설명으로 변환하여 text-to-video 모델에 필수적인 학습 데이터를 제공해야 함
고품질 동영상 캡션 데이터를 생성하기 위해 Dense Video Caption Data Generation 파이프라인을 구축

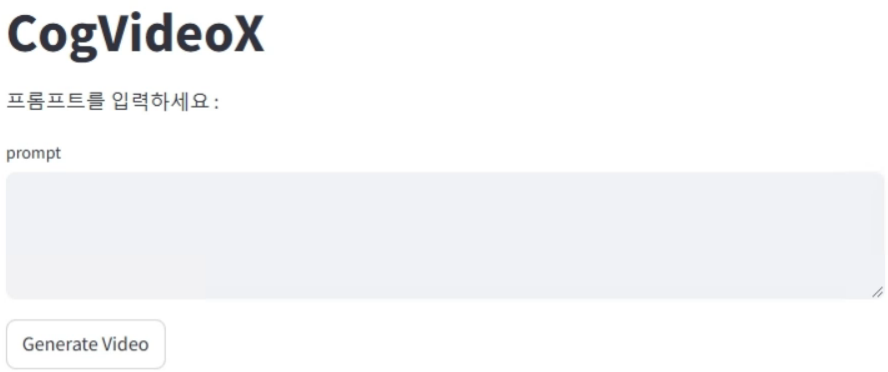


1. 동영상 captioning 모델인 Panda70M을 사용하여 동영상에 대한 짧은 캡션을 생성
2. 이미지 recaptioning 모델 CogVLM을 사용하여 동영상 내 각 프레임에 대한 고밀도 이미지 캡션을 만듦
3. GPT-4를 사용하여 모든 이미지 캡션을 요약하여 최종 동영상 캡션을 생성
4. 이미지 캡션에서 동영상 캡션으로의 생성을 가속화하기 위해 GPT-4에서 생성된 요약 데이터를 사용하여 Llama 2 모델을 fine-tuning하여 대규모 동영상 캡션 데이터 생성을 가능하게 함

코드 구현

```
import torch
from diffusers import CogVideoXPipeline
from diffusers.utils import export_to_video

prompt = "A cat, dressed in a small, red jacket and a tiny hat, sits on a wooden stool in a serene forest. \
    The cat's fluffy paws strum a miniature acoustic guitar, producing soft, melodic tunes. Nearby, a few other cats gather, \
        watching curiously and some clapping in rhythm. Sunlight filters through the tall tree, casting a gentle glow on the scene. \
            The cat's face is expressive, showing concentration and joy as it plays. The background includes a small, flowing stream and \
                vibrant green foliage,enhancing the peaceful and magical atmosphere of this unique musical performance."

pipe = CogVideoXPipeline.from_pretrained(
    "THUDM/CogVideoX-5b",
    torch_dtype=torch.bfloat16
)

pipe.enable_model_cpu_offload()
pipe.vae.enable_tiling()

video = pipe(
    prompt=prompt,
    num_videos_per_prompt=1,
    num_inference_steps=50,
    num_frames=49,
    guidance_scale=6,
    generator=torch.Generator(device="cuda").manual_seed(42),
).frames[0]

export_to_video(video, "./output_cat2.mp4", fps=8)
```

코드 구현



Thank You
Q & A