Generative Image Dynamics

Authors : Zhengqi Li, Richard Tucker, Noah Snavely, Aleksander Holynski (Google)

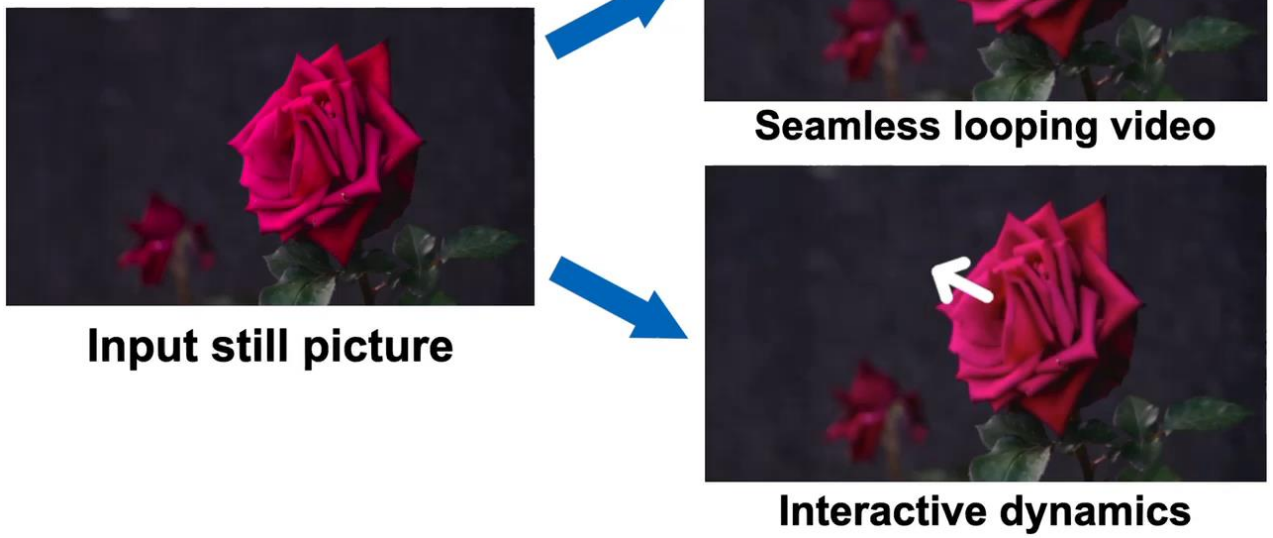
CVPR 2024 Best Paper Award

발표일자 : 2025.03.11 (화)

발표자 : 송건학

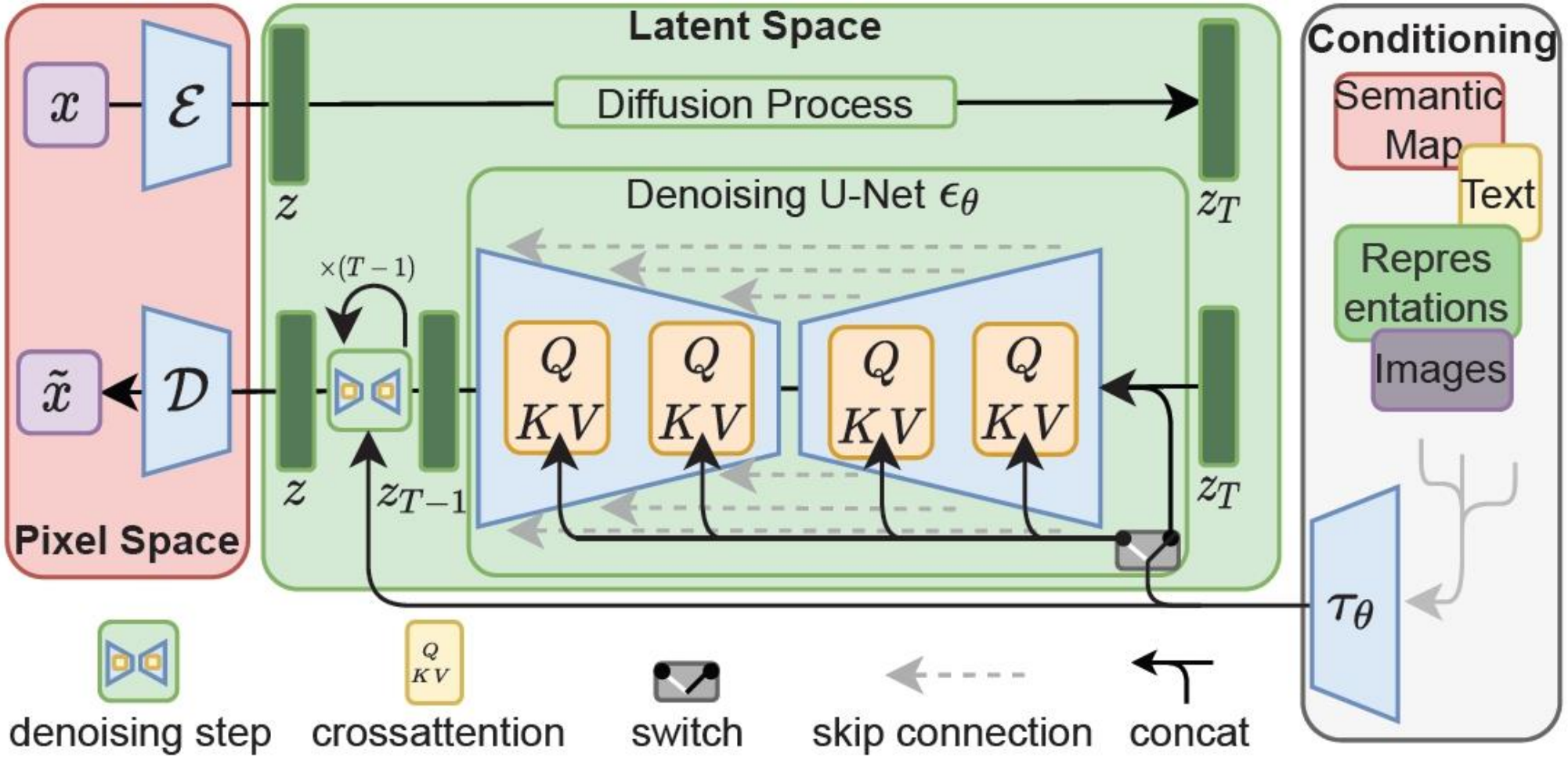
Table of Contents

Project Page Demo



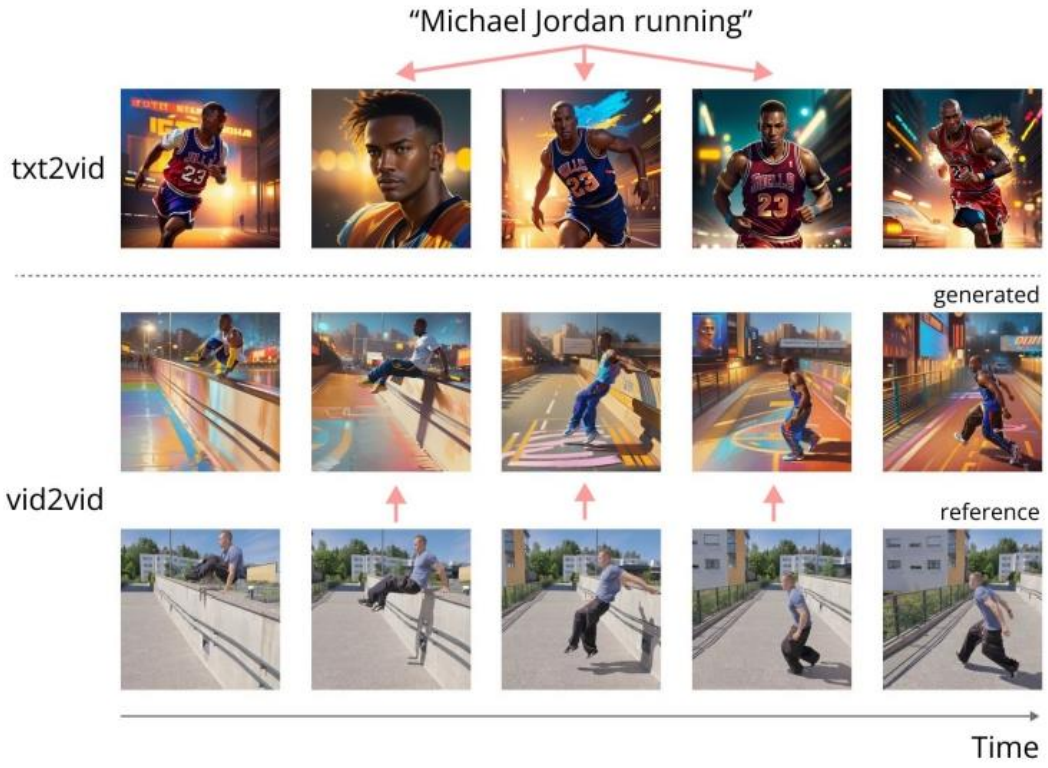
https://generative-dynamics.github.io

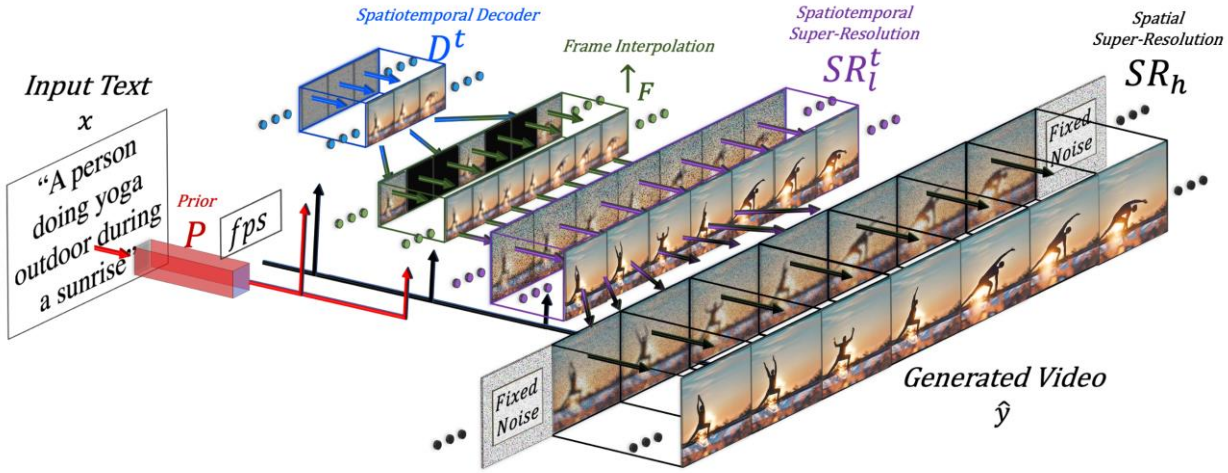
Preliminary – LDM (Stable Diffusion)

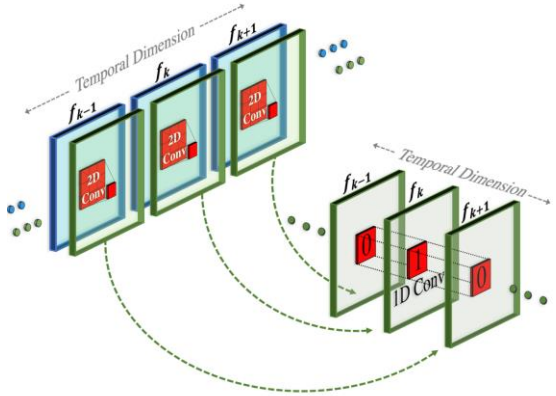


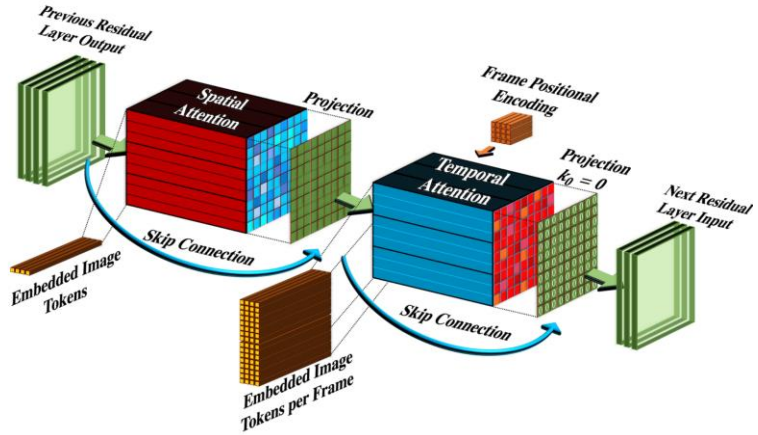
$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right]$$

Preliminary –Video Generation







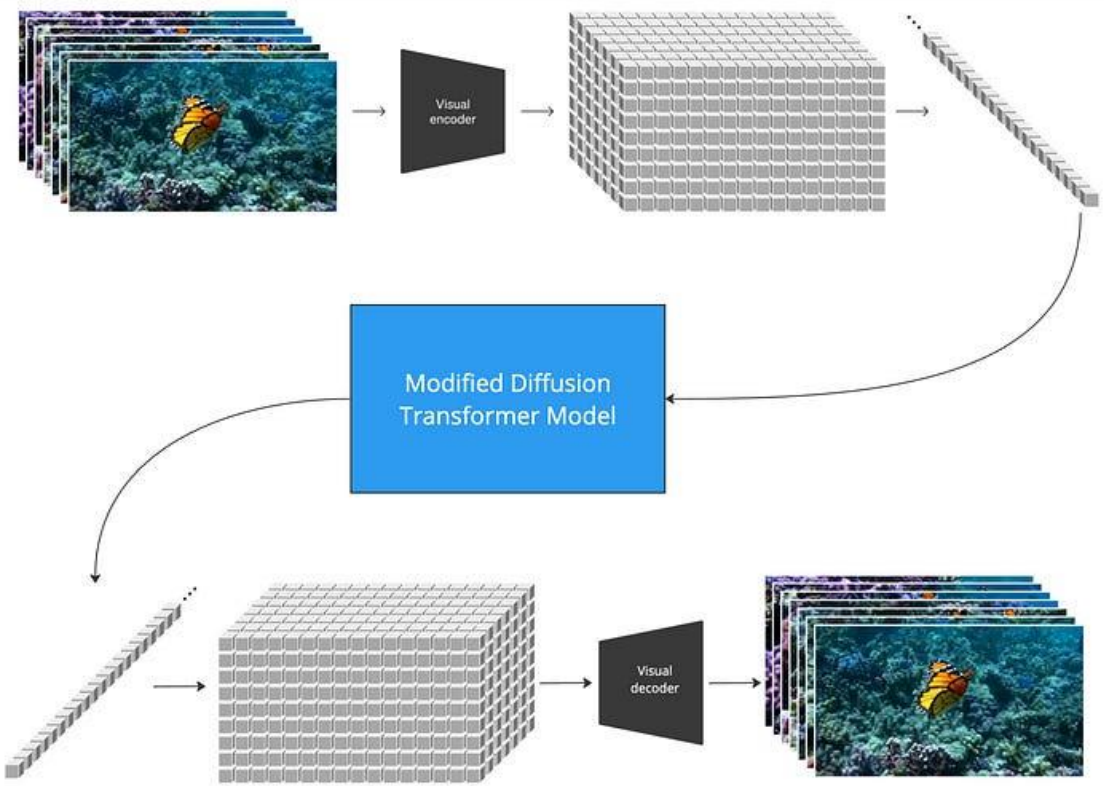


Video Diffusion Models: A Survey https://arxiv.org/abs/2405.03150

Make-A-Video https://arxiv.org/abs/2209.14792

Preliminary –Video Generation

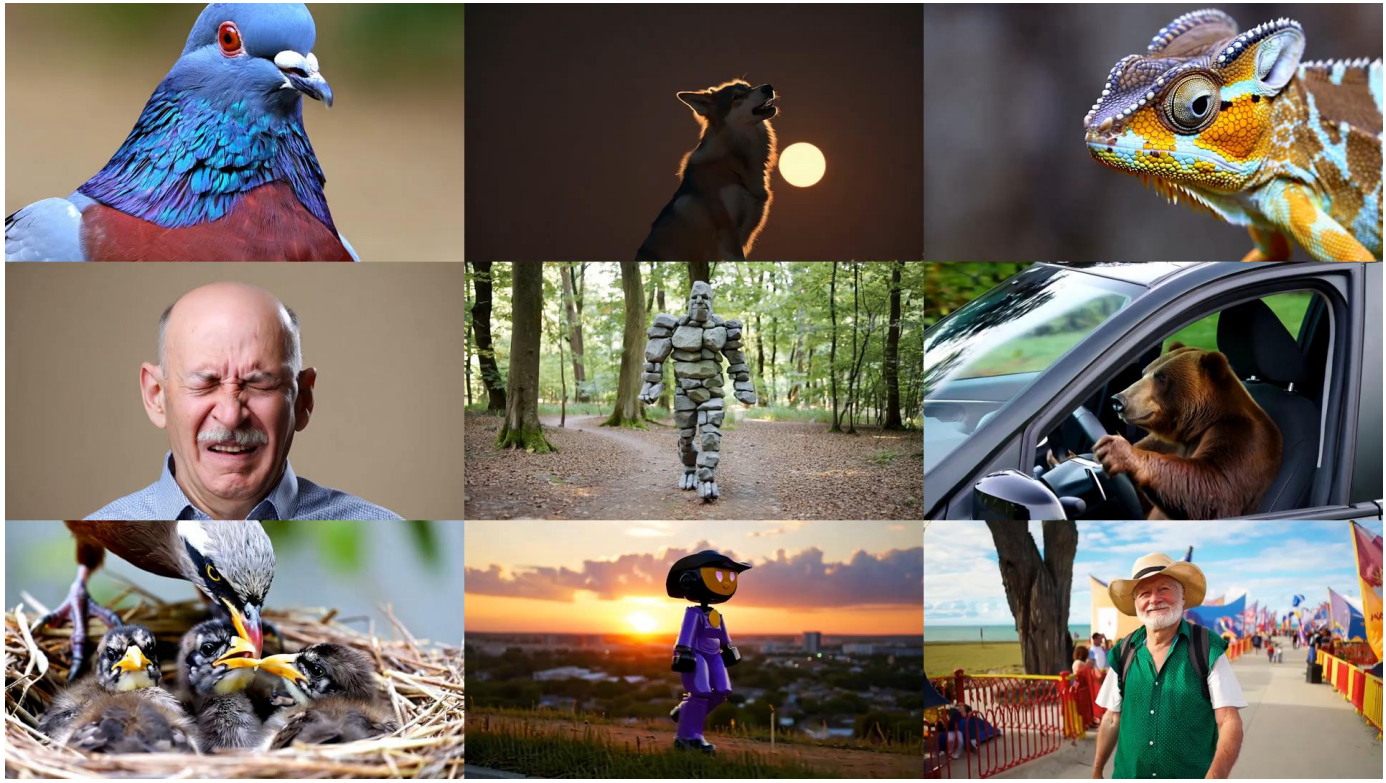
DiT (Diffusion Transformer) based Video Generation



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

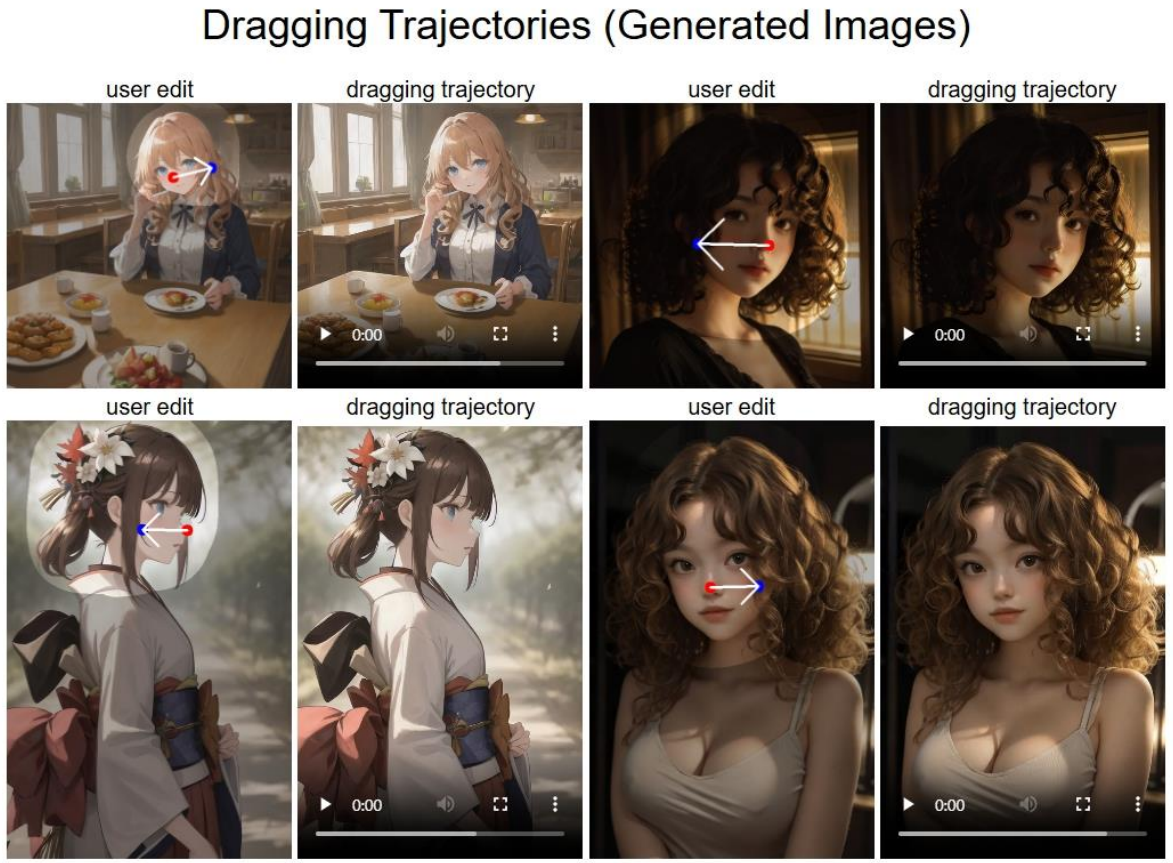
Flow Matching based Video Generation – Goku (25.02.07)

https://arxiv.org/abs/2502.04896



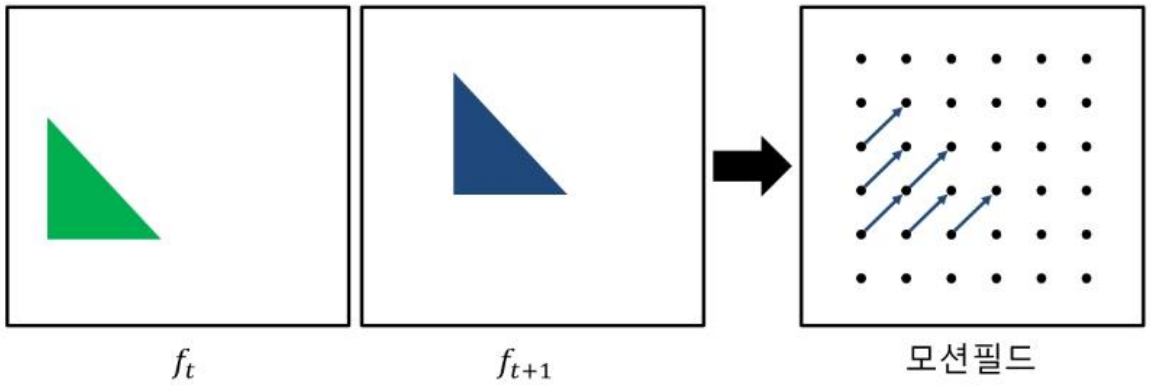
Preliminary – Point-based Manipulation Generative Image

DragGAN (SIGGRAPH 2023)
https://vcai.mpi-inf.mpg.de/projects/DragGAN/



DragDiffusion (CVPR 2024)
https://yujun-shi.github.io/projects/dragdiffusion.html

Preliminary – Optical Flow



https://gaussian37.github.io/vision-concept-optical_flow/

정의

1) 관찰자와 장면 사이의 상대적인 운동으로 인해 시각적 장면에서 물체, 표면 및 가장자리의 명백한 운동 패턴

2) 시간 축에 연속적인 이미지 데이터에서 각 픽셀의 위치를 추적하는 문제

(Motion Field = Optical Field = 움직임이 발생한 픽셀의 모션 벡터로 얻어낸 2차원 모션 맵)

알고리즘 : Lucas-Kanade, Horn-Schunck 알고리즘 (Classic CV Algorithm)

단점

1) 동영상 프레임 간 간격이 너무 크면 예측이 어려움

2) 조도와 같은 요인으로 픽셀 값 차이가 급격히 난다면 예측하기 어려움

Section 1. Introduction

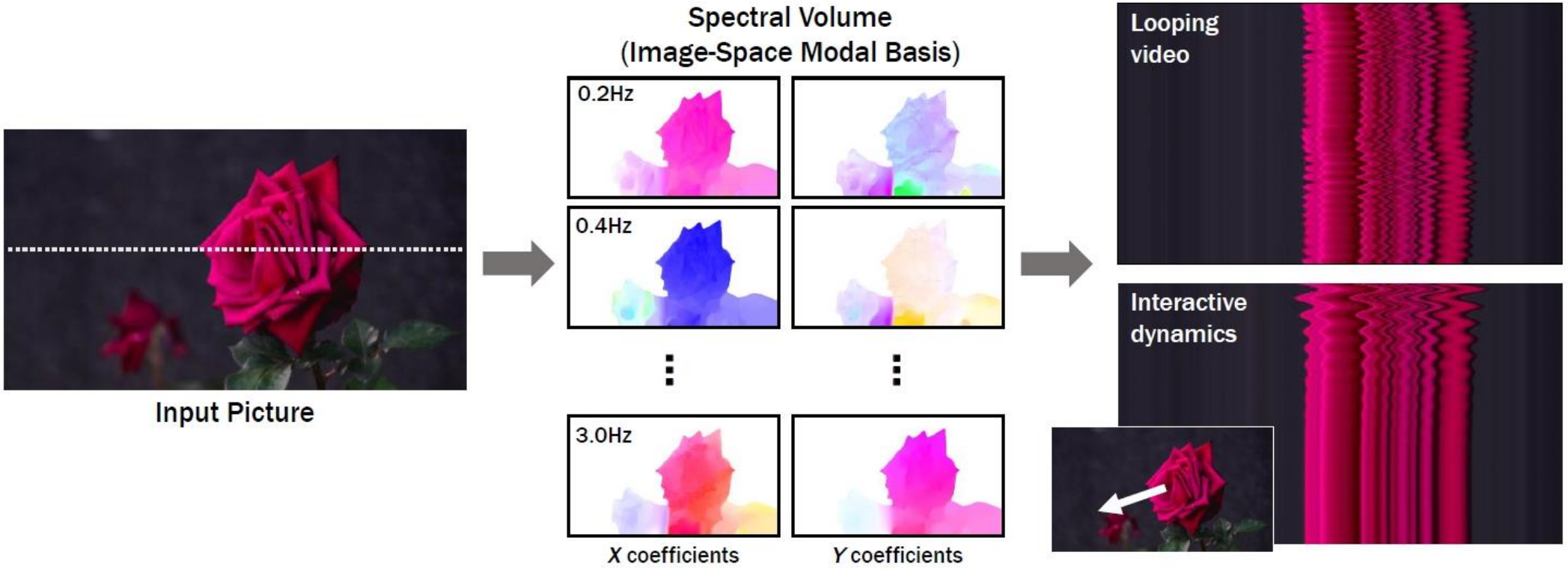
• 실제 세계는 정적인 장면처럼 보여도, 미세한 진동이나 바람·물결 등에 의해 모든 것이 움직인다.

• 사람은 단일 사진만 보고도 나뭇잎이 흔들리거나 촛불이 흔들리는 듯한 장면을 상상할 수 있지만, 컴퓨터가 그러한 사실적인 움직임을 학습하고 재현하기는 매우 어렵다.

• 다행히도, 실제 2D 영상 속 pixel trajectory만 충분히 파악해도, 물리적 속성(질량·탄성 등)을 직접 계산하지 않고도 그럴듯한 동적인 결과물을 생성할 수 있다. (2016 Visual vibration analysis)

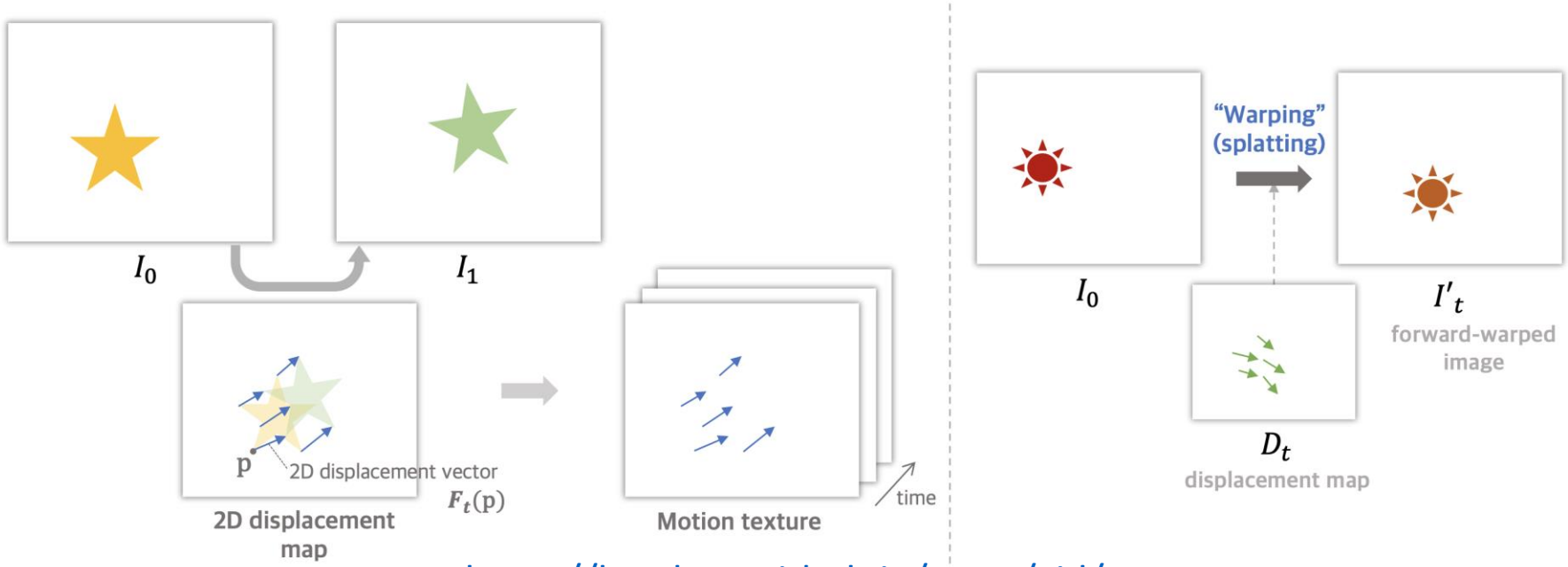
• 본 논문은 풍부한 데이터 분포를 보유한 **conditional diffusion model**을 활용하여 단일 이미지의 픽셀 단위 움직임(image-space scene motion)에 대한 generative prior 학습을 진행한다.

Section 1. Introduction



- 실제 영상에서 추출된 dense long-range motion trajectory을 **Spectral Volume**로 변환 진행
  - **Spectral Volume** 을 통해 frequency domain에서 주로 발생하는 자연스러운 **oscillatory dynamics motion을 효과적으로 모델링**.
- 예측된 Spectral volume을 motion texture로 변환한 후 이를 기반으로 animation을 제공함.

Section 1. Introduction



https://lunaleee.github.io/posts/gid/

- Motion Texture는 이미지의 각 픽셀에 대한 움직임 벡터를 시간 순서대로 나열한 것.
- 각 픽셀은 특정 시점에서 특정 방향으로 얼마나 이동하는지에 대한 정보를 담고 있으며,

이러한 정보는 2D 변위 맵 형태(2D displacement map)로 표현함.

- 이러한 변위 맵들을 시간 순서대로 연결하면, 이미지 전체의 움직임을 나타내는 Motion Texture가 됨.

Section 2. Related Work

Generative synthesis & Animating images

- 사실적인 이미지를 생성하는 텍스트-이미지 모델을 확장하여 비디오 시퀀스를 생성하는 연구들이 진행 중. 이미지 텐서를 시간 temporal dim으로 확장하는 방식을 사용. 하지만 이러한 방법들은 생성된 비디오에서 움직임이 부자연스럽거나 텍스처가 일관성 없이 변화하거나 물리적 제약(예: 질량 보존)을 위반하는 문제점 생김.
- 한계 극복을 위해 외부 source(video, motion, 3d geometry priors, annotations)를 기반한 motion generation을 진행하나 motion field를 통한 이미지 animating 또한 일관성 뛰어난 영상을 만들지만 추가적인 guidance나 움직임 표현에 제약 존재.

Motion models and motion priors

- **oscillatory 3D motion** 방식. 즉, Fourier domain에서 노이즈를 조작하여 시간 영역의 모션 필드를 생성하는 방식과 시스템의 역학을 분석하는 modal analysis에 기반한 접근 방식들 제안됨. Visual vibration analysis에 가장 큰 영감을 받음. 특히, frequency-space spectral volume motion representation 차용.
- 최근 **optical flow motion estimate**에 근간한 prediction task를 통해 motion representation도 활발히 진행됨.

Videos as textures

- **Dynamic Texture (Video Texture)** 움직이는 장면(예: 물결, 나뭇잎, 불꽃)을 시공간적 확률 분포로 간주하고, 그 패턴을 반복·연장해 무한히 재생 가능한 영상 제작함. 하지만 이러한 방법들은 대체로 동영상이 제공되거나, 사용자 지정(영상 프레임을 어떻게 연결할지 결정)에 대한 의존도가 높아, **임의의 정지 이미지에 적용**하기가 쉽지 않음

Section 3. Overview

- Input : 이미지 I_0
- Goal : Oscillatory Motion이 가능한 video 생성
- Pipeline : 2개의 Module (Motion Prediction Module, Image-based Rendering Module)

방법

1) Spectral Volume $\mathcal{S} = (S_{f_0}, S_{f_1}, S_{f_2}, \cdots, S_{f_{K-1}})$ 예측을 위한 LDM 사용
2) 예측한 spectral volume은 Inverse Discrete Fourier Transform을 통해 motion texture $\mathcal{F} = (F_1, F_2, \cdots, F_T)$로 변환.
이 motion은 모든 future time step에서 각 입력 픽셀의 위치를 결정함.
3) Motion Texutre가 주어졌을 때, image-based Rendering 기술을 활용하여 RGB 이미지에 애니메이션 적용

Section 4.1 Motion Representation

- Motion Texture : 입력 이미지의 픽셀 좌표 p에서의 2D displacement vector $F_t(p)$ sequence

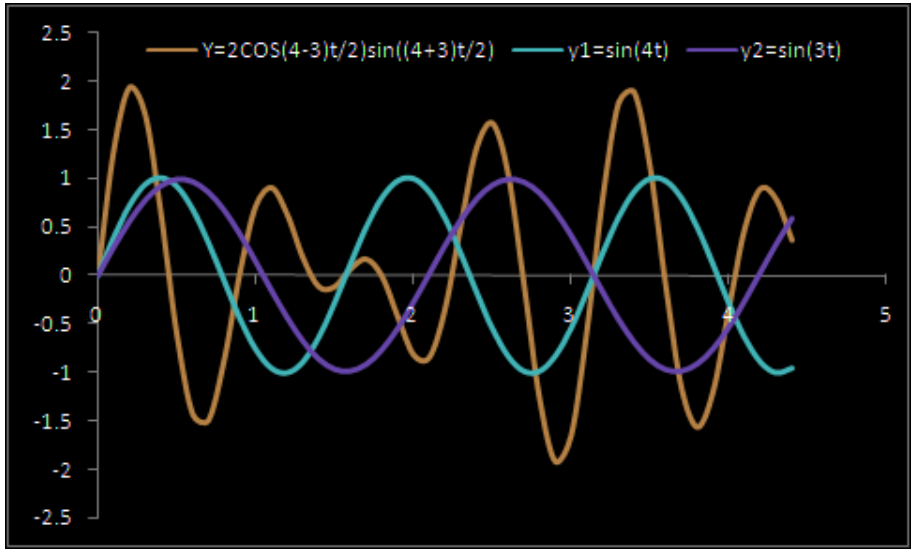
a forward-warped image I'_t $\quad\quad$ $I'_t(\mathbf{p} + F_t(\mathbf{p})) = I_0(\mathbf{p}).$

- 목표에 부합하는 비디오 생성을 위해서는 Motion texture이 반드시 필요하다.

- 단순히 motion texture로 video를 생성할 수 있으나 video 길이에 따른 motion texture의 사이즈도 커짐.
  (e.g. T개의 output frame을 위해서 T 개의 displacement field가 필요)
- 긴 비디오 생성을 위한 많은 output 생성을 회피하기 위한 2가지 방법
  (Autoregressively video frame을 생성, 추가 time embedding을 통해 독립적으로 예측)을 수행해도
  long-term temporal consistency에 대한 제약이 발생함.

- **즉, 직접적으로 Motion Texture를 생성하는 방식은 한계점이 존재한다.**

Section 4.1 Motion Representation



왜 Spectral Volume을 활용하는가

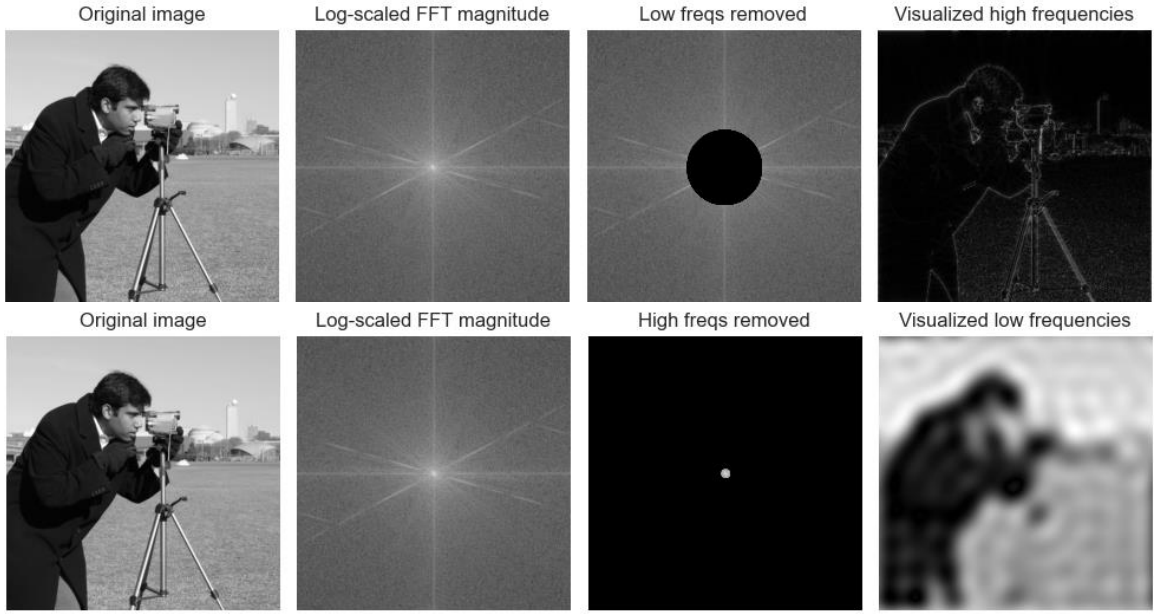
- 다행히도 많은 natural motion은 각각 다른 frequency, amplitude, phase을 가지는 **harmonic oscillation의 중첩(superposition)**으로 표현 가능.
- 이러한 motion은 quasi-periodic.
  즉, 준주기적 (완전히 반복적이지는 않지만 일정한 패턴을 보이는 움직임)이므로
  frequency domain에서 모델링하는 것이 유리.
- **Spectral Volume[**Davis et al.]은 **비디오에서 움직임을 주파수 영역으로 표현하는 효율적인 방법**

Section 4.1 Motion Representation

- 입력 : Image → 출력 : motion Spectral volume
- LDM을 활용하여 4K-channel 2D motion spectral map으로 구성된 Spectral volume 생성
- K << T는 모델링된 frequency 숫자. 4개의 scalar 구성(x, y 차원 Complex Fourier Coefficients)

$$\mathcal{S}(\mathrm{p}) = \mathrm{FFT}(\mathcal{F}(\mathrm{p})).$$

- Spectral Volume
  = Fast Fourier Transform (Motion Trajectory)

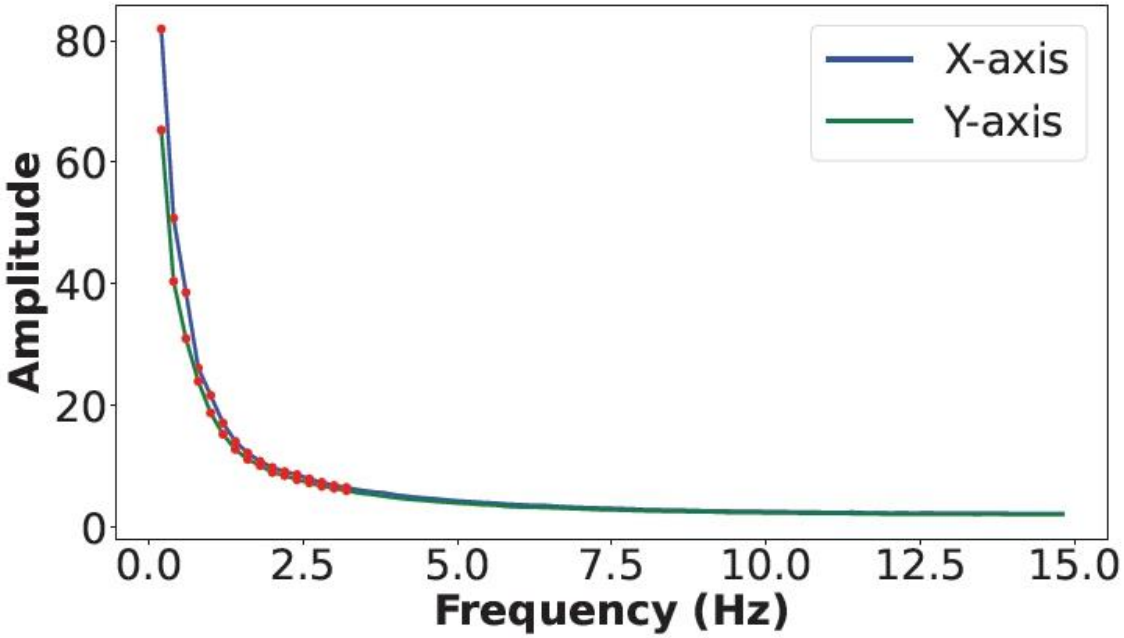


참조

- Fourier Transform은 시간에 따라 변화하는 신호(예: 픽셀의 움직임)를 다양한 주파수 성분으로 분해하여 분석하는 방법 (e.g. image compression, restoration etc)

https://towardsdatascience.com/the-fourier-transform-4-putting-the-fft-to-work-38dd84dc814/

Section 4.1 Motion Representation

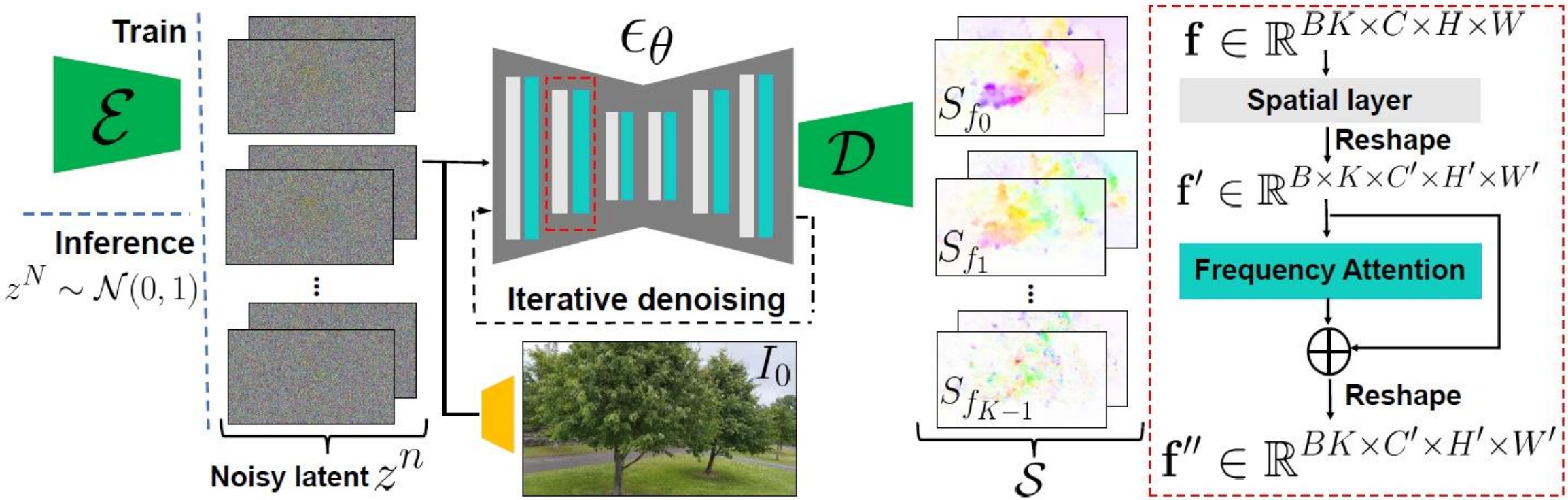


Frequency K의 개수 선택

- 이전 연구 :
  자연스러운 진동 움직임이 주로 low frequency 성분으로 구성됨을 확인.

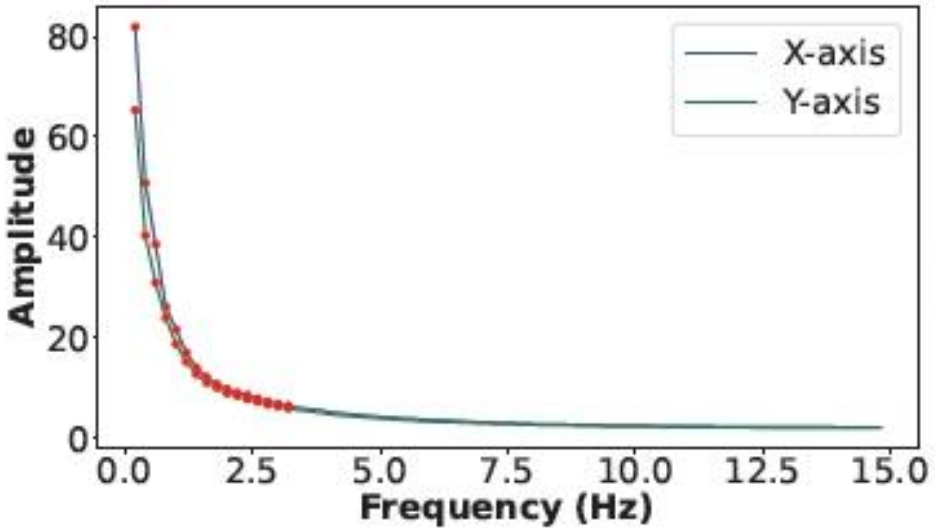
- 실제 비디오 클립에서
  추출한 움직임의 평균 Power Spectrum 분석
- 주파수가 증가함에 따라 Amplitude 지수적으로 감소
- K=16으로도 비디오에서 자연스러운 움직임.

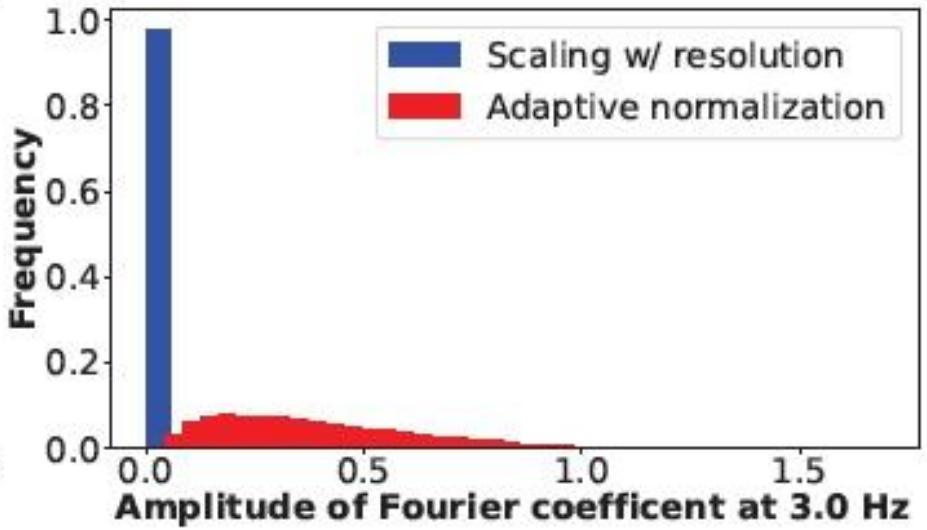
Section 4-2. Predicting motion with a diffusion model



$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{n\in\mathcal{U}[1,N],\epsilon^n\in\mathcal{N}(0,1)}\left[||\epsilon^n - \epsilon_\theta(z^n; n, c)||^2\right] \quad (3)$$

Section 4-2. Frequency adaptive normalization





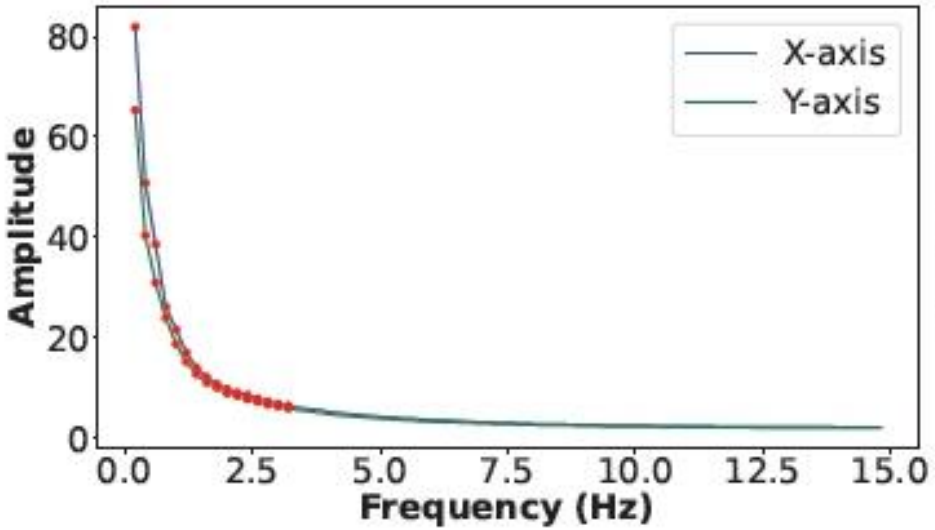
문제점:

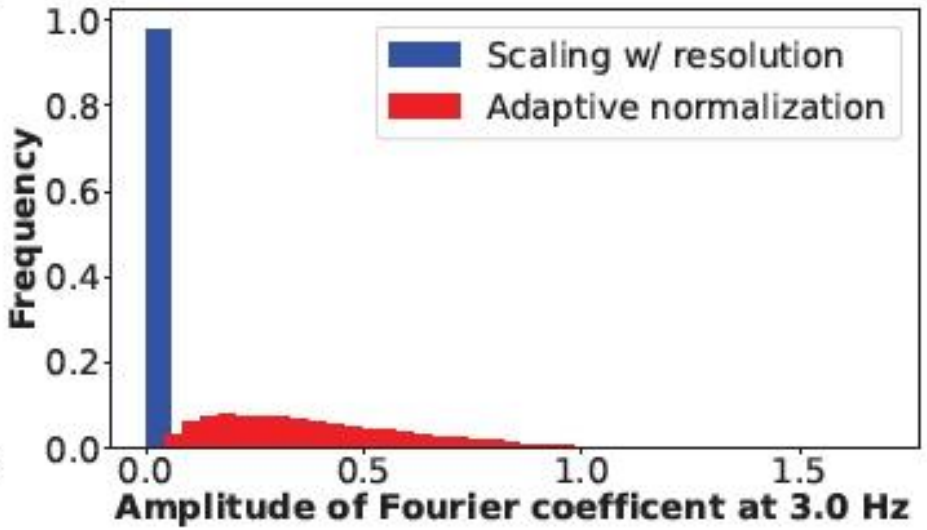
1) Spectral Volume의 진폭은 0~100이며 주파수가 증가함에 따라 지수적으로 감소

2) diffusion model은 안정적인 학습과 출력값이 -1~1 사이가 되도록 normalize할 필요가 있음

3) coefficient를 0~1로 만들 경우 고주파 성분이 대부분 0에 가까워 예측 오차가 크게 나타날 수 있음.

Section 4-2. Frequency adaptive normalization





해결책 순서

$$S'_{f_j}(\mathbf{p}) = \text{sign}(S_{f_j}) \sqrt{\left| \frac{S_{f_j}(\mathbf{p})}{s_{f_j}} \right|}.$$

1) **Independently normalize Fourier coefficient at each frequency**:

각 주파수 f_j에 대해, 훈련 세트에서 95th percentile를 기반으로 Fourier 계수를 독립적으로 정규화.

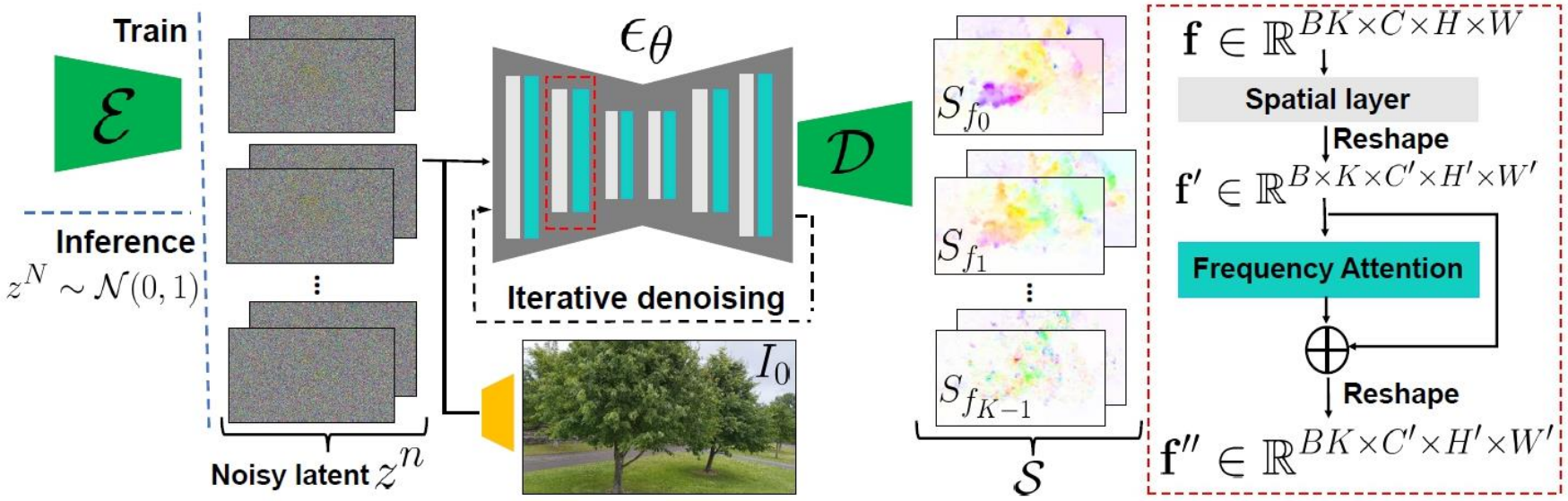
이 값을 **주파수별 스케일링 요소 s_{f_j}**로 사용.

2) **Power Transformation**:

스케일링된 Fourier 계수에 power Transformation을 적용하여 값이 **극단적인 값으로 치우치지 않도록** 함.

실험적으로 **제곱근 변환**이 로그나 역수와 같은 다른 비선형 변환보다 성능이 더 좋습니다.

Section 4-2. Frequency-coordinated denoising

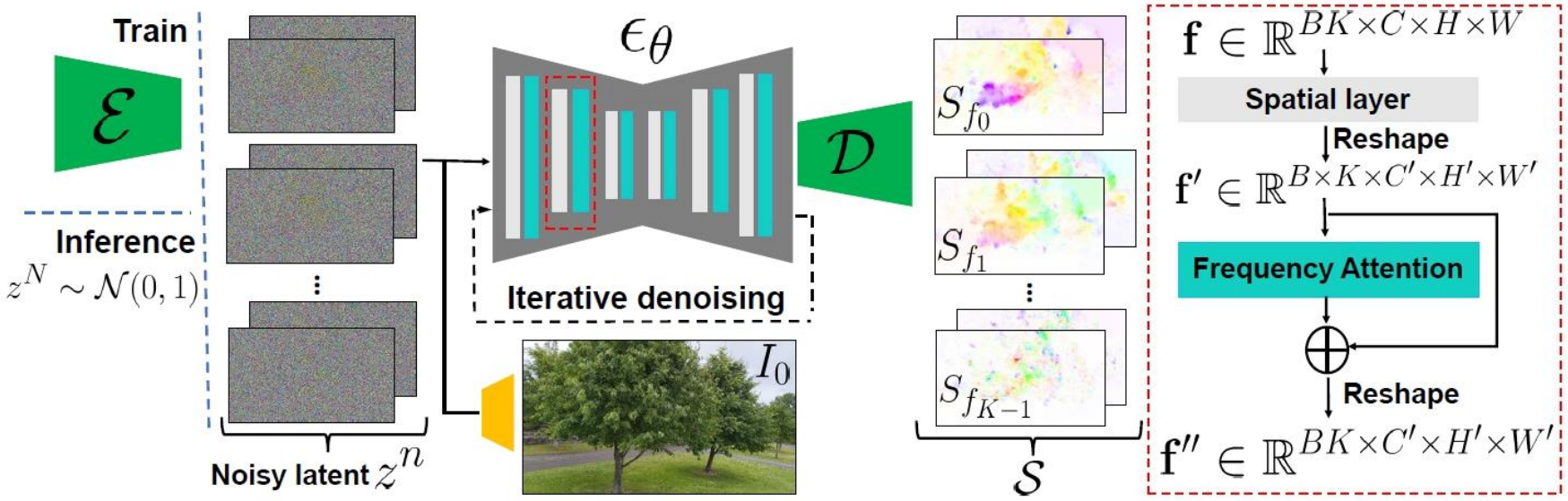


문제점:

1) **K frequency band의 spectral volume 예측시 4K Channel tensor 출력하는 경우**:
많은 채널 생성으로 인해 결과물이 **흐릿해지고 부정확**해질 수 있다는 이전 연구 결과 존재.

2) **frequency embedding 주입을 통해 각 frequency slice를 독립적으로 예측시:**
frequency 영역에서 상관 관계가 없는 예측을 초래하여 **부자연스러운 움직임이 생성**될 수 있다.

Section 4-2. Frequency-coordinated denoising

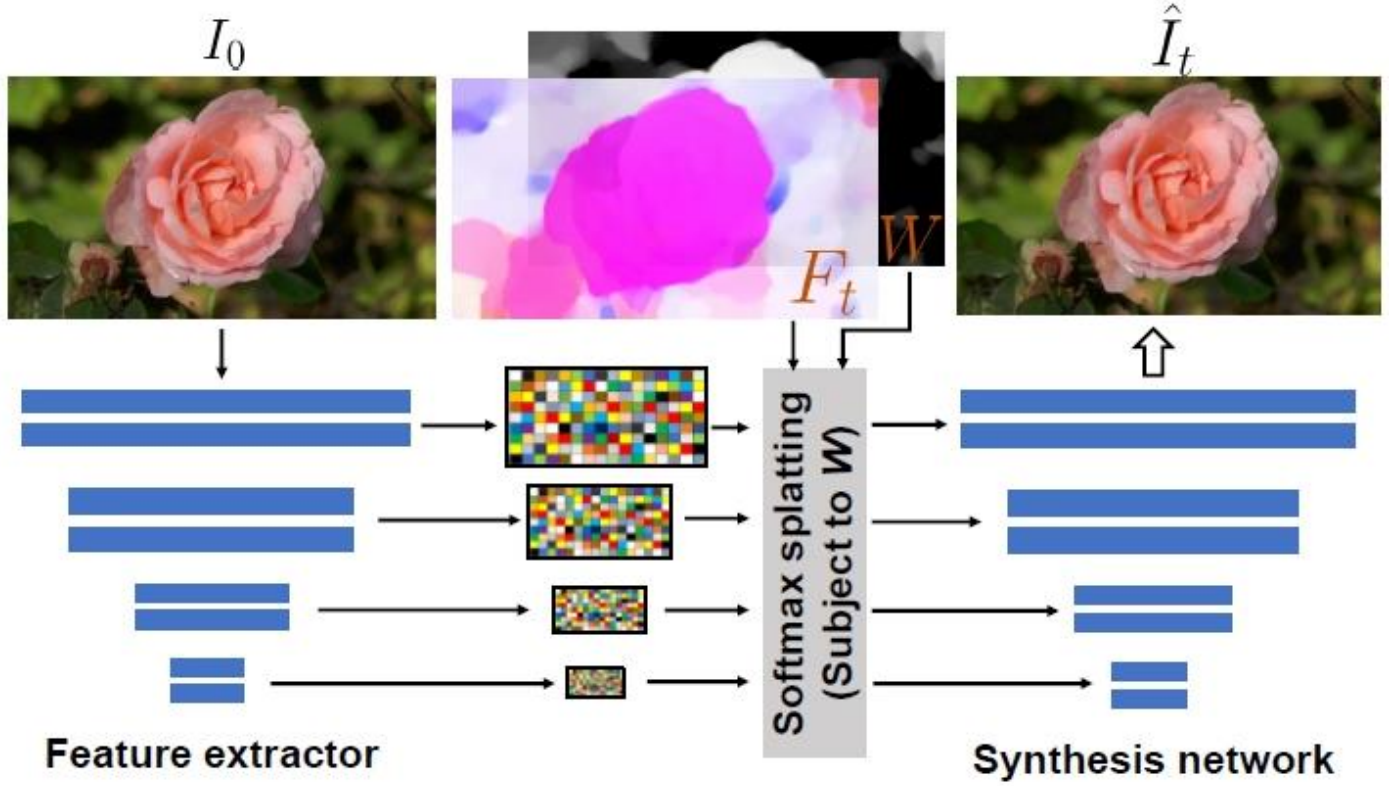


제안 전략: Frequency-coordinated denoising strategy

순서 1) LDM(Latent Diffusion Model)을 학습시켜 Spectral volume S_{f_j} 의 **단일 4채널 frequency slice를 예측**.
이때, time-step embedding을 따라 추가적인 frequency embedding을 LDM에 주입. (주파수들간 상호작용 고려 X)
순서 2) 학습된 LDM의 파라미터 freeze

순서 3) 2D spatial layer와 **attention layer**를 K frequency band에 걸쳐 교차로 배치(interleave)한 후 fine-tuning.
이를 통해 frequency attention layer가 **모든 frequency slice**를 조정하여 **일관성 있는 Spectral volume 생성**

Section 5. Image-based rendering



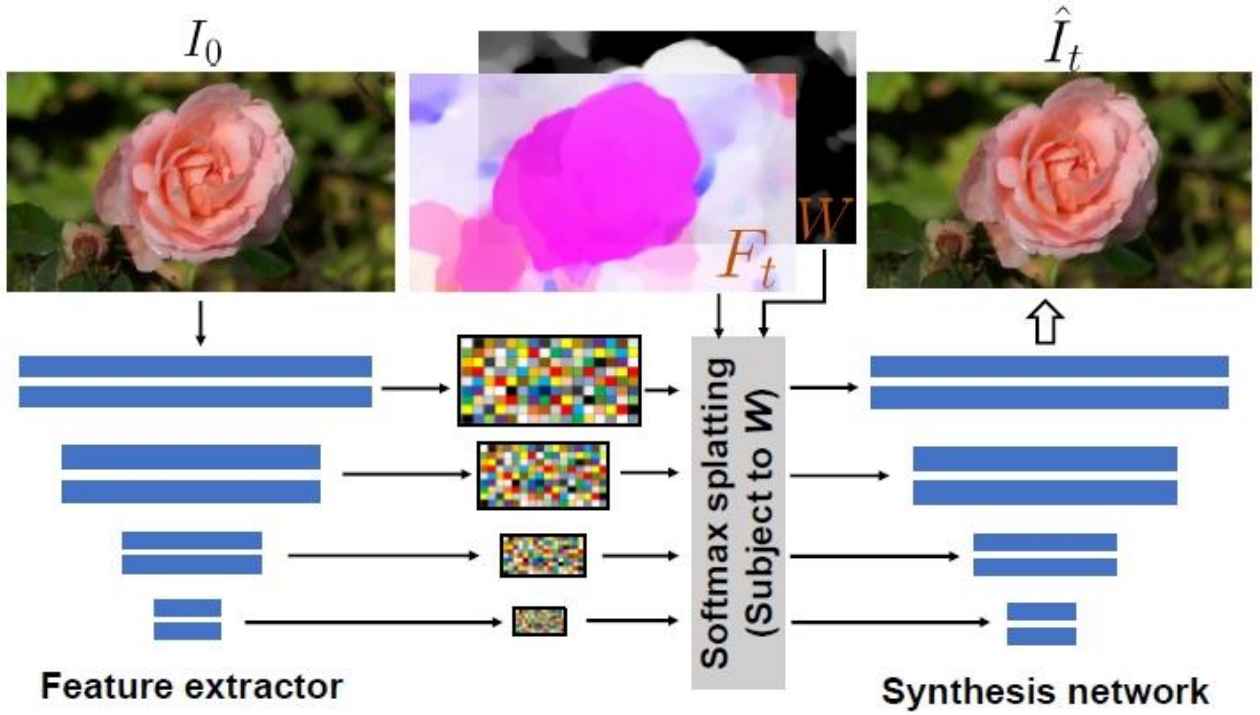
입력 이미지 I_0에서 예측된 Spectral volume S을 사용하여 미래 시점 t에서의 frame $\hat{I}_t$을 render 방법 소개.

순서 1) 픽셀별 Inverse temporal FFT를 적용하여 time domain motion texture $\mathcal{F}(p)$를 얻습니다.

$$\mathcal{F}(p) = \mathrm{FFT}^{-1}(S(p))$$

순서 2) Deep Image-based Rendering Technique (**Feature Pyramid Softmax Splatting Strategy** 수행)

Section 5. Image-based rendering



Feature Pyramid Softmax Splatting Strategy

순서 1) Feature Extractor Network를 통한 입력 이미지의 multi-scale feature map 추출

순서 2) Scale j에 따른 해상도별 Motion Field F_t resize & scale

순서 3) 예측된 Flow Magnitude를 Depth Proxy로 사용하여 각 소스 픽셀의 가중치 W 결정

$$W(p) = \frac{1}{T}\sum_t ||F_t(p)||^2$$

순서 4) Softmax Splatting 적용:
Motion field F_t와 가중치 W 를 사용하여 각 스케일의 Feature map에 **softmax splatting**을 적용하여 warped feature를 생성.

순서 5) Image Synthesis Decoder에 Warped Feature 주입하여 최종 rendering 이미지 생성

Experiments

Implementation details

- Spectral Volume을 위해 LDM backbone (2D U-Net, MSE loss)
- VAE L1 reconstruction loss, multi-scale gradient consistency loss, KL-divergence loss를 사용하여 학습
- 256x160 크기의 이미지 / 16개의 Nvidia A100 GPU / 6일 학습 / 최대 512x288 해상도 비디오 생성
- IBR (Image-Based Rendering) 모듈에서 feature 추출기로 ResNet-34가 사용
- Nvidia V100 GPU에서 실시간으로 25FPS로 실행

Data

- Oscillatory Motions이 보이는 3,015개 온라인 비디오 수집 및 자체적으로 촬영 (10% 테스트용)
- 최종 15만 개 이상의 이미지-모션 쌍으로 구성된 데이터 세트를 구축

Experiments

Method	Image Synthesis FID	KID	Video Synthesis FVD	FVD_{32}	DTFVD	$DTFVD_{32}$
TATS [35]	65.8	1.67	265.6	419.6	22.6	40.7
Stochastic I2V [27]	68.3	3.12	253.5	320.9	16.7	41.7
MCVD [93]	63.4	2.97	208.6	270.4	19.5	53.9
LFDM [67]	47.6	1.70	187.5	254.3	13.0	45.6
DMVFN [48]	37.9	1.09	206.5	316.3	11.2	54.5
Endo *et al.* [29]	10.4	0.19	166.0	231.6	5.35	65.1
Holynski *et al.* [46]	11.2	0.20	179.0	253.7	7.23	46.8
Ours	**4.03**	**0.08**	**47.1**	**62.9**	**2.53**	**6.75**

Table 1. **Quantitative comparisons on the test set.** We report both image synthesis and video synthesis quality. Here, KID is scaled by 100. Lower is better for all error. See Sec. 7.1 for descriptions of baselines and error metrics.

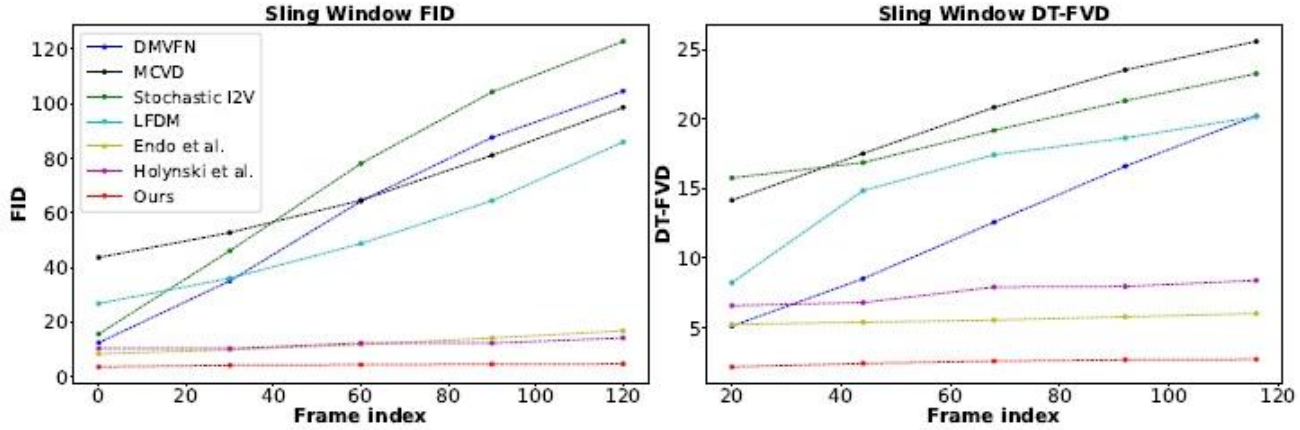


Figure 6. **Sliding window FID and DTFVD.** We show sliding window FID with window size 30 frames, and DTFVD with size 16 frames, for videos generated by different methods.

[비디오의 품질이 시간에 따라 어떻게 변하는지 측정]

시간 경과에 따라 더 일관되고 현실적인 비디오를 생성

Experiments

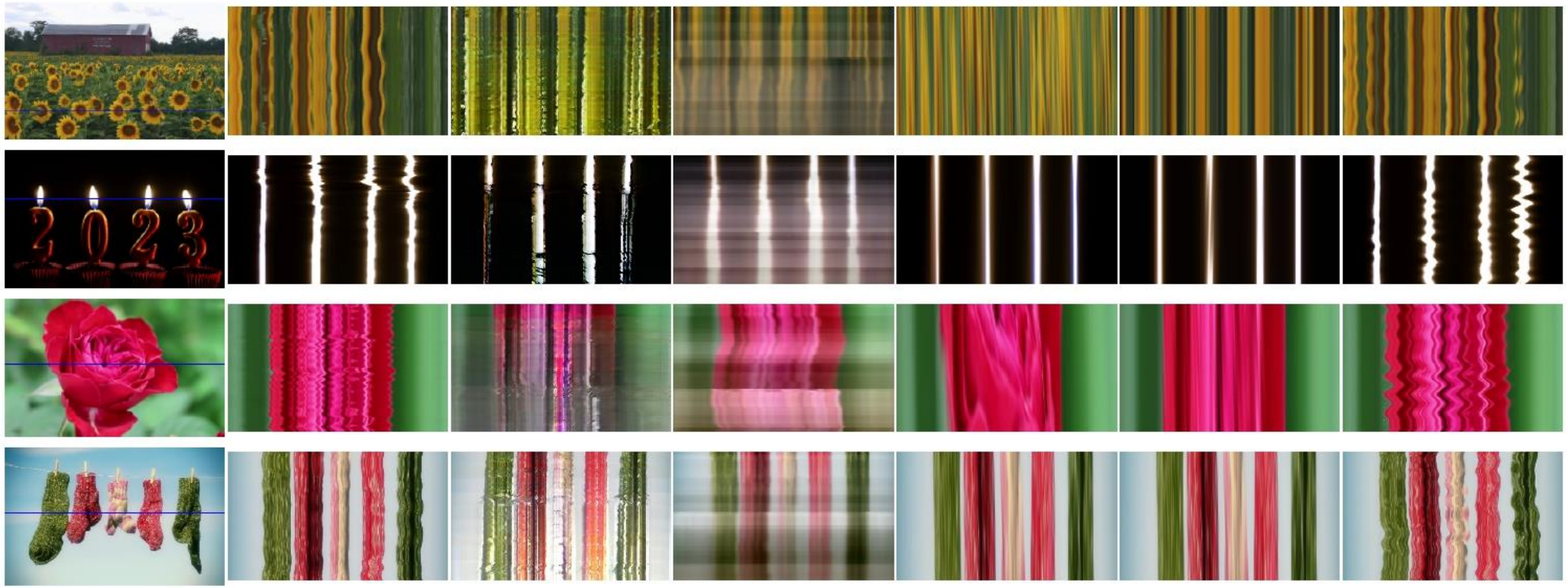


Figure 5. X-t **slices of videos generated by different approaches.** From left to right: input image and corresponding X-t video slices from the ground truth video, from videos generated by three baselines [27, 29, 46, 93], and finally videos generated by our approach.

비디오의 공간적 차원(x축)과 시간적 차원(t축)을 결합하여 비디오 내 움직임을 시각적으로 표현하는 방법

Experiments & Limitation

Method	Image Synthesis FID	KID	Video Synthesis FVD	FVD_{32}	DTFVD	$DTFVD_{32}$
Repeat I_0	-	-	237.5	316.7	5.30	45.6
$K = 4$	3.92	0.07	60.3	78.4	3.12	8.59
$K = 8$	3.95	0.07	52.1	68.7	2.71	7.37
$K = 24$	4.09	0.08	48.2	65.1	2.50	6.94
w/o adaptive norm.	4.53	0.09	62.7	80.1	3.16	8.19
Independent pred.	4.00	0.08	52.5	71.3	2.70	7.40
Volume pred.	4.74	0.09	53.7	71.1	2.83	7.79
Baseline splat [46]	4.25	0.09	49.5	66.8	2.83	7.27
Full ($K = 16$)	4.03	0.08	47.1	62.9	2.53	6.75

Table 2. **Ablation study.** Sec. 7.3 describes each configuration.

Figure 7. We show generated future frames from three recent large video diffusion models [31, 36, 98].

Figure 8. **Limitations.** We show examples of rendered future frames (even), and overlay of input and rendered images (odd). Our method can produce artifacts in regions of thin objects or large motions, and regions requiring filling large amount of new contents.

한계 1) spectral volume의 low frequency만 예측하기 때문에, non-oscillating motion이나 high frequency 모델링 실패.

한계 2) 생성된 video의 품질은 underlying motion trajectories의 품질에 의존하며, 이는 얇은 moving objects 또는 큰 displacement를 가진 objects가 있는 장면에서 저하

Why “Generative Image Dynamics” won the Best Paper Award

기술적 혁신성 & 독창성:

- **기존 단일 이미지 기반 비디오 생성 모델들과의 차별점:** 본 논문은 단일 이미지 기반의 Latent Diffusion Model(LDM) 활용하여 Spectral Volume이라는 주파수 도메인 표현을 고안하였고, 이에 기반한 natural motion에서 발생하는 oscillatory 동작을 효율적으로 모델링함.
- **주파수 도메인 처리의 장점:** low frequency 기반 특성을 세밀하게 조정·추론하는 frequency adaptive normalization이나 frequency-coordinated denoising 등의 기법을 제공함으로써 길이가 긴 동영상에서도 일관적이고 안정적인 움직임을 구현함에 독창성이 존재

기술적 완성도:

- **높은 품질** : 정교한 주파수 기반 접근과, 긴 시간 스케일에 대한 충실도 높은 동영상 생성 성능에 높은 점수

재현성 :

- **구체적인 학습 세팅과 리소스 명시**: 논문에서 평균 GPU 사용량, hyperparameter 등 **세부 기술 사항이 명시**

활용성 :

- **폭넓은 응용 분야:** Image-to-video, Seamless looping, 단일 이미지 기반 Interactive dynamics animation 가능.
  그래픽/엔터테인먼트 분야에 잠재적 응용 가능성이 존재
- **실시간 처리 가능성:** 25 FPS까지 달성된다고 언급된 부분을 통해 상호작용 애플리케이션에서 실질적으로 사용 가능