『2026 하계 세미나』

Integrating Video Diffusion into 4D Scene Modeling



Pusan National University
Dept. of Electronics Engineering
Computer Vision & Signal Processing Lab.



Presented By
최지훈

Background

- Gaussian Splatting
  - 가우시안으로 표현된 다수의 타원체들의 집합
    - 3D공간 상에 다수의 Gaussian 타원체들을 배치하여 3D 공간을 표현
      - 가우시안들은 3D 위치, 크기, 회전, 투명도, 색상 정보를 가짐
    - 원하는 카메라 View Point에서 가우시안들을 Image Plane으로 Projection하고 각 가우시안의 색상과 투명도를 누적하여 이미지를 렌더링



< 3D Gaussian Splatting >

Background

- Gaussian Splatting
  - 3D/4D Reconstruction
    - 3D공간을 표현하기 위해서 GT 이미지와 렌더링 이미지 사이의 Loss를 이용하여 Gaussian의 Parameter(위치, 투명도, 색상 등)를 반복적으로 최적화
      - 필요에 따라 Gaussian을 추가하거나 제거
    - 4D 복원의 경우 별도로 3D로 만든 Gaussian의 시간에 따른 변화도 함께 별도의 모듈을 통해 함께 학습
    - 최근에는 장면마다 반복적인 optimization을 수행하는 방식에서, neural network가 Gaussian parameters를 직접 예측하는 feed-forward reconstruction으로 확장되는 연구도 활발히 진행

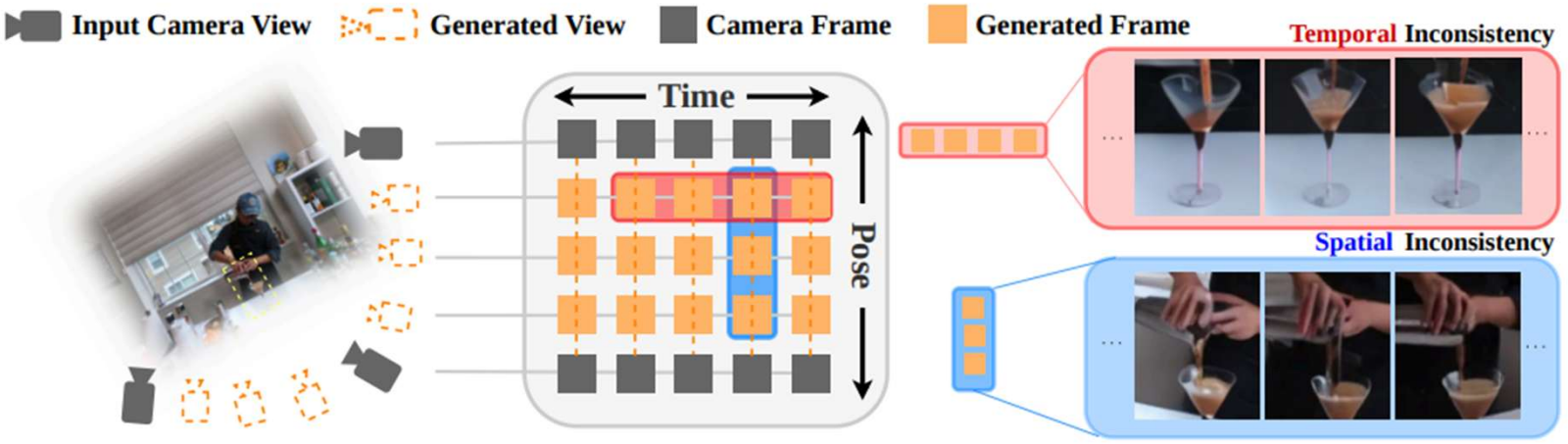


< 4DGS >

SparseCam4D: Spatio-Temporally Consistent 4D Reconstruction from Sparse Cameras [CVPR 26]

SparseCam4d

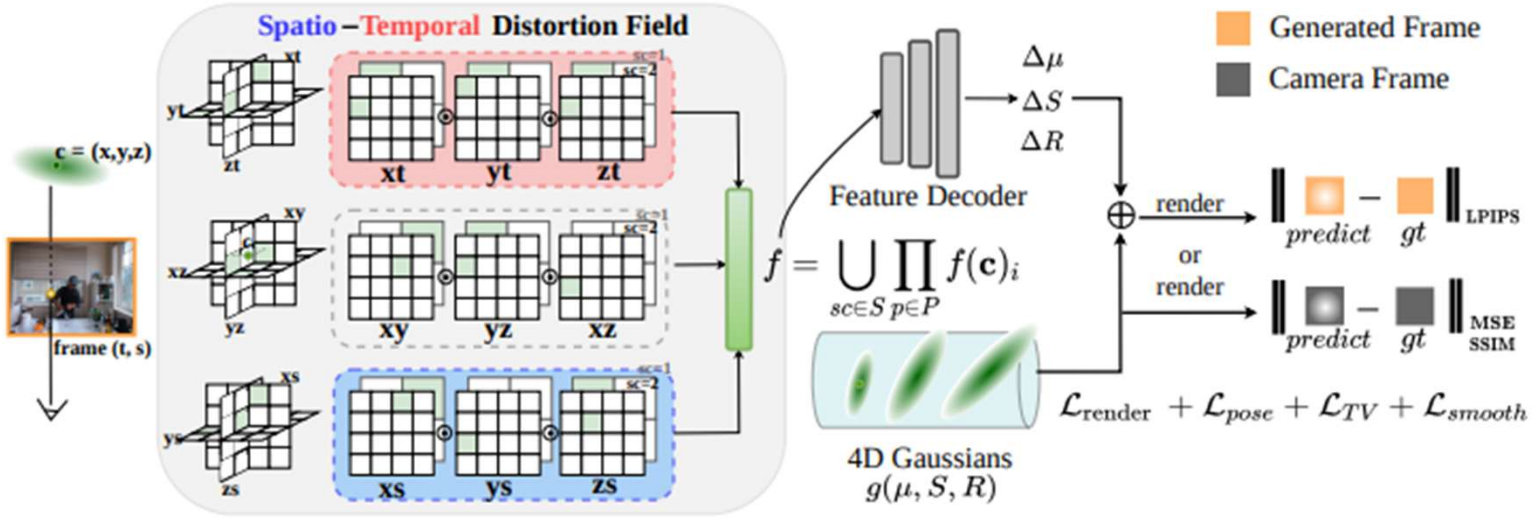
- Problem
  - Lack of Viewpoint
    - Dynamic 4D Scene Reconstructin은 충분한 다시점의 비디오가 필요로 하지만, 실제 환경에서는 사용할 수 있는 Camera 수가 적음
  - Distortion of Video Diffusion
    - 디퓨전 모델을 통해 생성한 이미지/비디오의 필연적인 Hallucination과 기하학적 불일치는 3D Reconstruction에 치명적



< Spatio-temporal inconsistency >

SparseCam4d

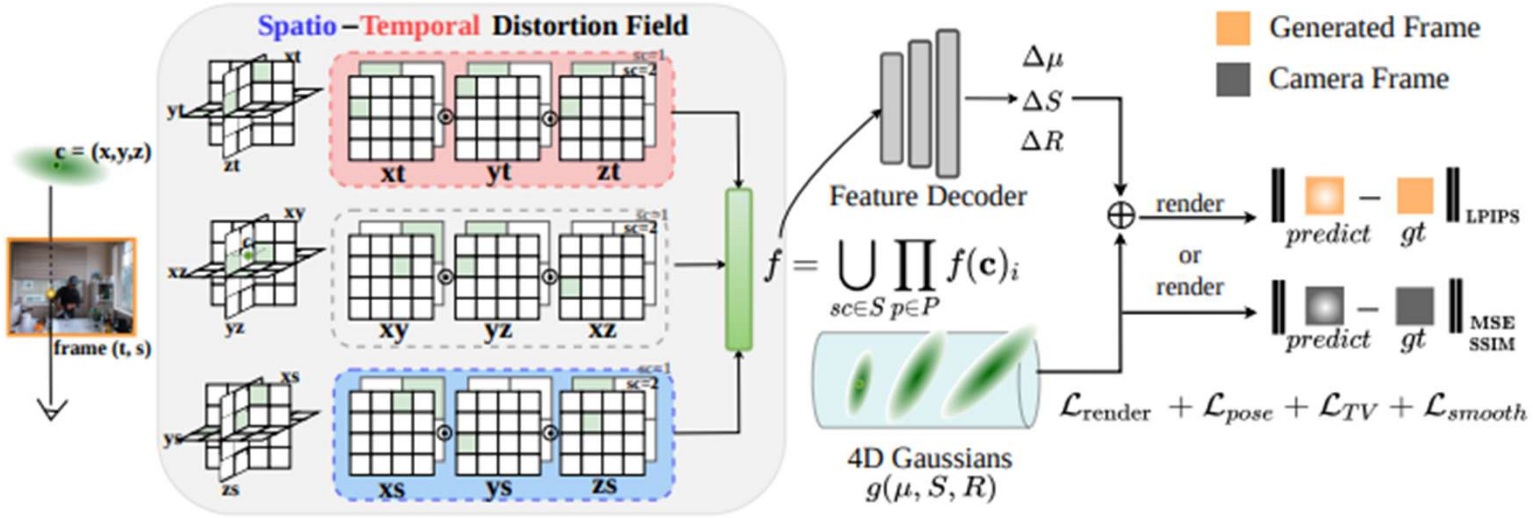
- Overview
  - Spatio-Temporal Distortion Field
    - 4DGS 학습과 동시에 생성된 pseudo-view의 시간적·공간적 왜곡을 모델링하는 별도의 Distortion Field를 학습
    - Pseudo-view는 Canonical 4D Gaussian에 Distortion을 적용하여 렌더링하고, 실제 관측 이미지는 Canonical 4D Gaussian만으로 렌더링



< Overview of Sparsecam4d >

SparseCam4d

- Overview
  - Spatio-Temporal Distortion Field
    - 이를 통해 Distortion Field는 pseudo-view에만 존재하는 비일관적인 변화를, Canonical 4D Gaussian은 여러 관측에서 공통적으로 나타나는 일관된 scene information을 학습
    - 최종 rendering에서는 Distortion Field를 제거하고 canonical 4D Gaussian representation만을 사용



< Overview of Sparsecam4d >

SparseCam4d

- Method
  - Generating Multiview Video
    - Multiview Images 생성 모델인 ViewCrafter를 이용하여 각 Time에서의 MultiView Images Set을 생성
    - 생성된 Image들을 시간축으로 연결하여 Multiview Video 생성



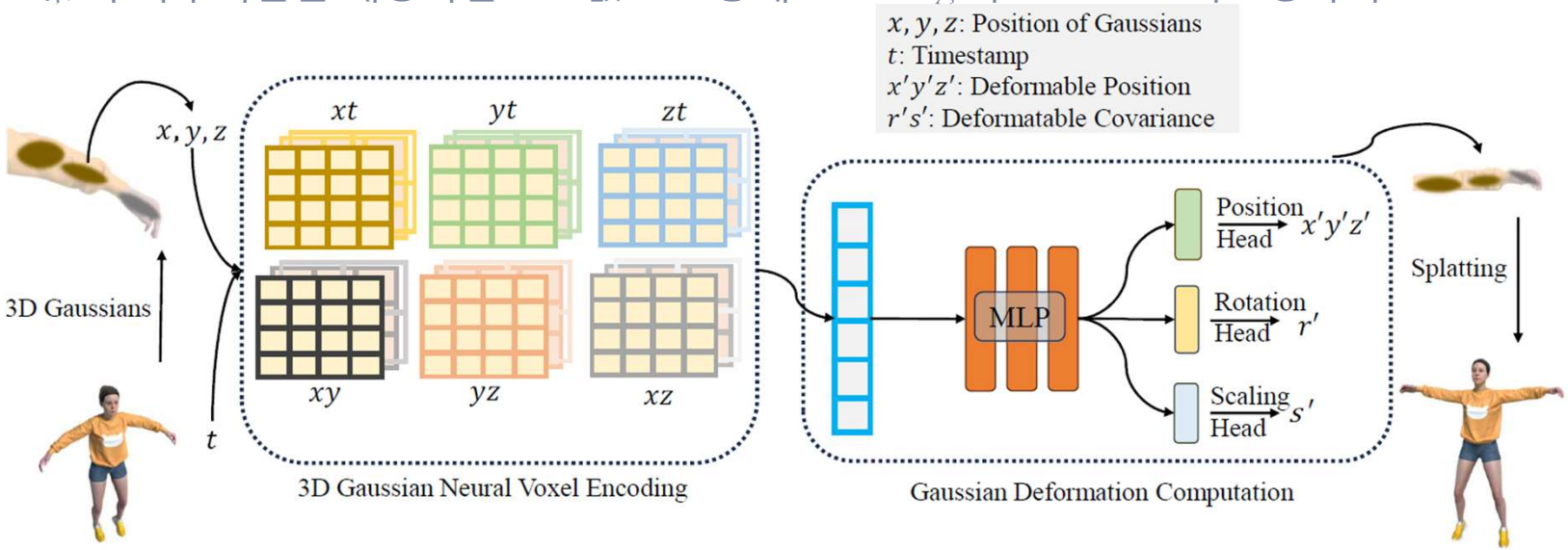
< Multiview Video >



< Temporal Video >

SparseCam4d

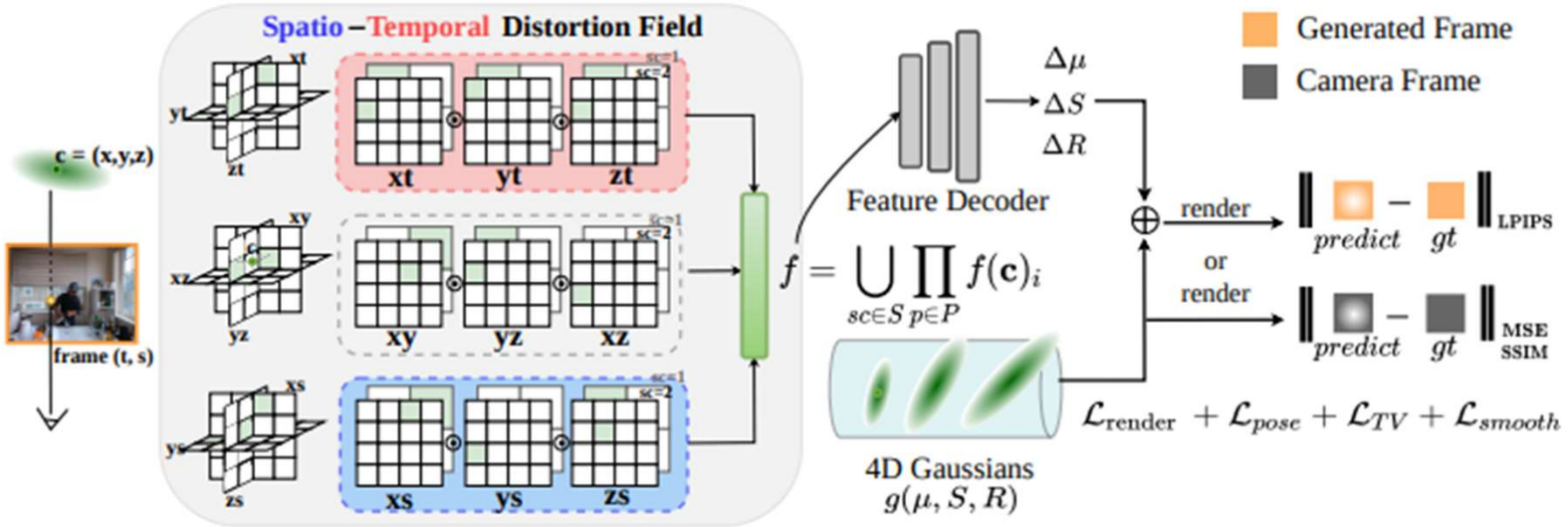
- Method
  - Distortion Field
    - Distortion의 deformation을 표현하기 위해 HexPlane-style factorized feature representation을 사용
    - HexPlane은 고차원 공간을 여러 개의 2D feature plane으로 factorization하여 표현하는 방식
      - x, y, z, t에 대하여 2차원 Grid들을 생성하고 Grid의 칸을 학습
      - 각 가우시안은 해당하는 Grid값으로 통해 Encoding 하고 MLP로 디코딩하여 Deformation



< Overview of Hexplane on 4DGS >

SparseCam4d

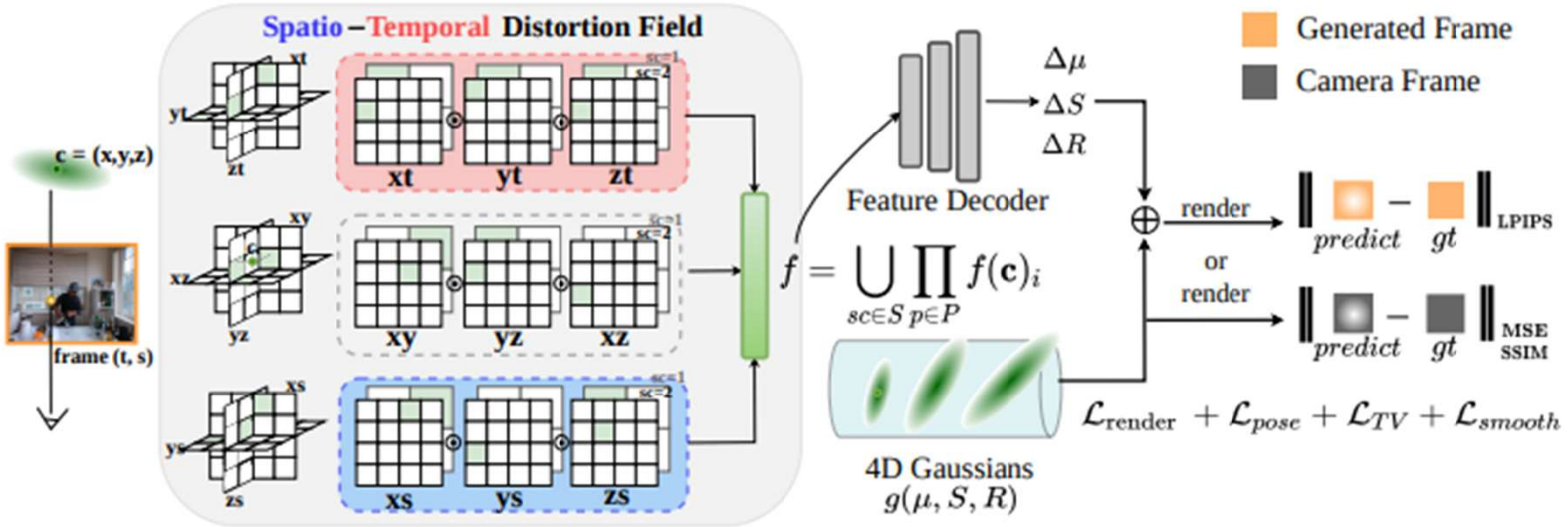
- Method
  - Distortion Field
    - Distortion Field는 기존 Hexplane에 View 축을 추가
      - x, y, z, t, s에 대하여 2차원 Grid들을 생성하고 Grid의 칸을 학습
      - 어떤 3D 위치의 Gaussian이 시간 또는 view에 따라 어떻게 달라지는지를 학습
      - t, s만으로는 공간상의 어느 가우시안이 왜곡되는지 표현할 수 없어 ts plane은 제외



< Overview of Sparsecam4d >

SparseCam4d

- Method
  - Training
    - 순수 Camera를 통해 얻은 GT 이미지는 Disotrtion Field를 사용하지 않고 가우시안을 렌더링하여 Loss 계산
    - 생성 이미지는 Distortion Field를 통해 Canonical 4D Gaussian에 변형을 주어 렌더링한 이미지와 Loss를 계산
      - Distortion Field에 의한 변형은 생성 이미지의 왜곡에 맞게 학습



< Overview of Sparsecam4d >

SparseCam4d

- Experiments
  - Quantitative Result
    - 3대의 카메라만으로 GS 학습 후의 정량적 결과
      - 기존 4D모델 대비 뚜렷한 성능 향상이 나타남

Method	Technicolor			Neural 3D Video			Nvidia Dynamic Scenes		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HyperReel [1]	14.14	0.453	0.616	15.63	0.582	0.500	19.88	0.528	0.396
4DGaussians [38]	16.20	0.505	0.552	17.40	0.673	0.320	16.81	0.372	0.516
4D-Rotor [5]	14.85	0.426	0.581	18.20	0.708	0.357	19.38	0.508	0.389
RealTime4DGS [43]	16.53	0.510	0.542	17.31	0.649	0.442	17.91	0.479	0.426
MonoFusion* [35]	17.97	0.578	0.352	18.43	0.738	0.270	20.22	0.590	0.192
Ours	23.15	0.728	0.299	21.91	0.789	0.258	24.81	0.794	0.150

< Quantitative Reuslts >

PSNR: 픽셀 단위 영상 복원 품질

SSIM: 밝기 대비 구조 기반 영상 유사도

LPIPS: 딥러닝 Feature 기반 영상 차이

SparseCam4d

- Experiments
  - Qualitive Result



< Video Reuslts >

SparseCam4d

- Experiments
  - Ablation Study
    - Distortion Field 적용시 PSNR 등 정량적 성능 향상이 나타남
    - 기존 생성모델 ViewCrafter 뿐만 아니라 ReCamMaster에서도 Distortion Field의 성능이 입증됨

Cook Spinach	PSNR↑	SSIM↑	LPIPS↓
ViewCrafter (w/o STDF)	21.42	0.775	0.302
ViewCrafter (w STDF)	23.93	0.832	0.232
ReCamMaster (w/o STDF)	21.97	0.756	0.315
ReCamMaster (w STDF)	23.61	0.806	0.247

< Ablation Reuslts >

PSNR: 픽셀 단위 영상 복원 품질

SSIM: 밝기 대비 구조 기반 영상 유사도

LPIPS: 딥러닝 Feature 기반 영상 차이

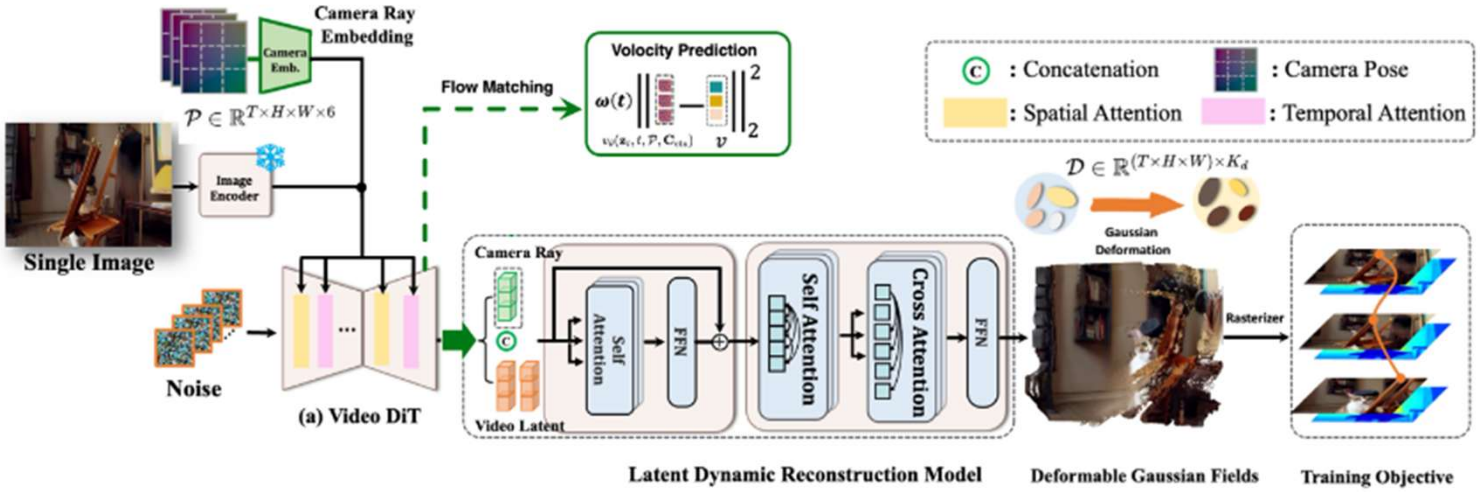
Diff4Splat: Repurposing Video Diffusion Models for Dynamic Scene Generation [CVPR 26]

Diff4Splat

- Motivation
  - Problem
    - Multi-stage 방식
      - Image to Video Generation 후 4D Reconstruction
      - 여러 단계를 거치며 추가적인 Reconstruction cost가 발생하고, 생성 단계의 오류가 후속 reconstruction에 전달될 수 있음
    - Direct feed forward 방식
      - 기존 Feedforward 방식은 주로 2D Video나 static 3D 머물러 있을 뿐, 4D Representation을 생성하지는 못함
  - Solution
    - Video Diffusion의 Generative Prior를 latent 그대로 활용하면서, single image로 부터 Dynamic 3D Scene Representation을 Feedforward로 직접 생성

Diff4Splat

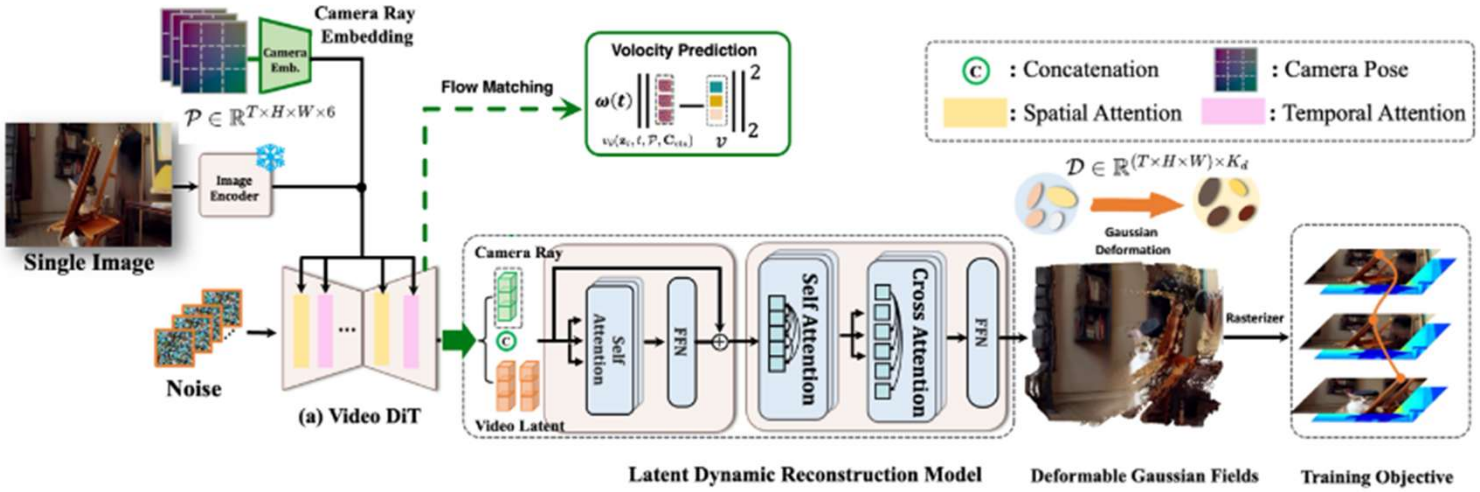
- Method
  - Overview
    - Camera-conditioned Video Diffusion으로 Video latent를 생성
    - LDRM(Latent Dynamic Reconstruction Model)이 Video Latent와 Camera 정보를 이용해 Feed Forward로 Gaussian representation 생성



< Overview of Diff4Splat >

Diff4Splat

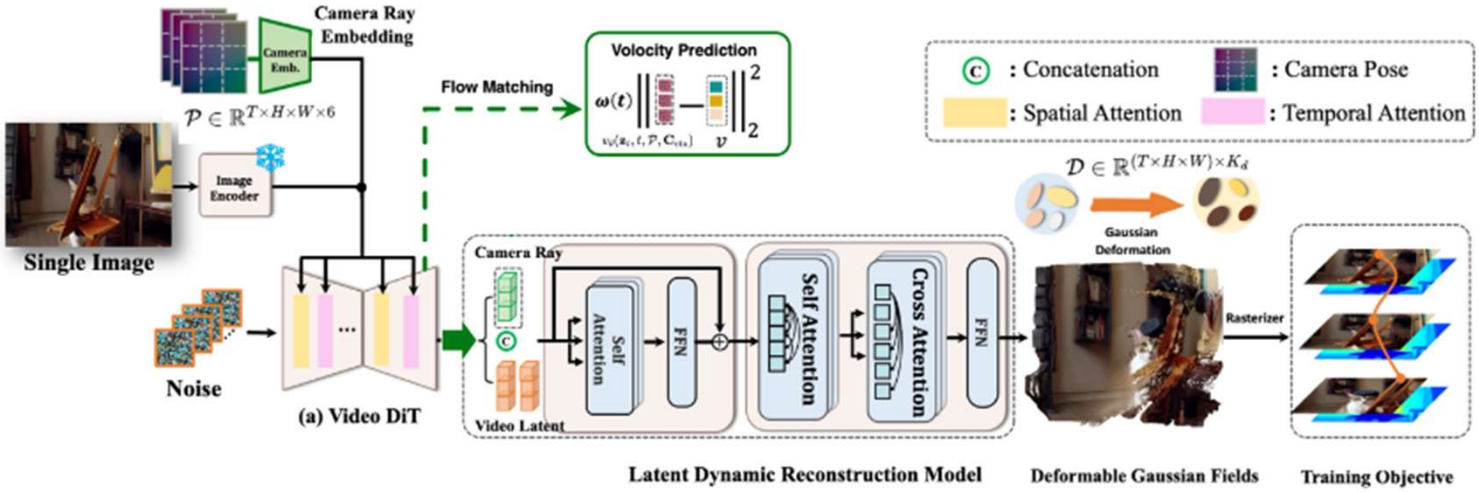
- Method
  - Video Diffusion as Generative Prior
    - Camera-conditioned Video Diffusion으로 Video latent를 생성
    - Video Diffusion은 단일 이미지에서 관측되지 않은 appearance와 dynamic motion을 생성할 수 있는 prior를 제공



< Overview of Diff4Splat >

Diff4Splat

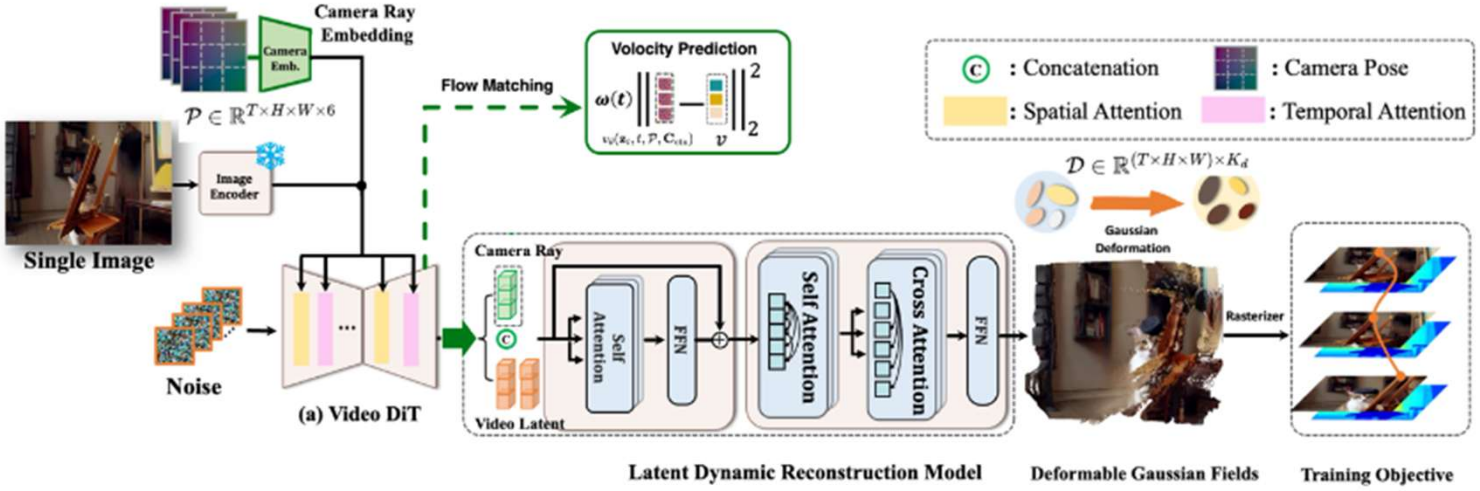
- Method
  - LDRM(Latent Dynamic Reconstruction Model) – Canonical Gaussian Prediction
    - Latent Token과 Camera Pose의 Pose Token을 결합하여 Trnaformer로 처리
    - Latent의 spatial/temporal 위치에 대응되는 gaussian parameter를 dense하게 예측
      - Latent로부터 (T, H, W, C)의 비디오를 디코딩 하듯이 (T, H, W, K_g)의 Gaussian Feature Map 예측
        - ✓K_g는 Gaussian Parameters



< Overview of Diff4Splat >

Diff4Splat

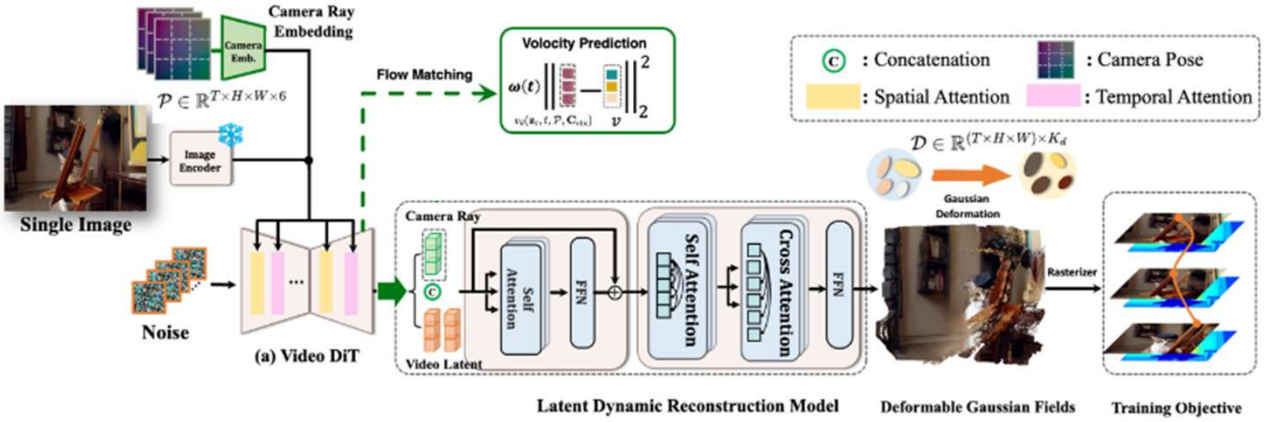
- Method
  - LDRM(Latent Dynamic Reconstruction Model) – Modeling Dynamic Scene
    - Canonical Gaussian + Time-dependent Deformation
      - 앞서 예측한 Canonical Gaussian을 장면의 기본 3D representation으로 사용
      - 각 Target Time t에 대하여 각 Canonical Gaussian에 상응하는 별도의 Position / Rotation / Scale 변화량을 예측
      - 각 시간의 3D Scene을 독립적으로 생성하는 대신, 하나의 Canonical Scene + 시간에 따른 변화량으로 전체 Dynamic Scene을 표현
        - ✓(T, H, W, K_d)의 Deformation Map / K_d는 Deformation Parameter



< Overview of Diff4Splat >

Diff4Splat

- Training
  - Flow Matching Loss
    - GT Video Latent를 이용하여 Pre-trained Video Diffusion Model을 Fine-tuning
    - Video Diffusion의 Latent Space를 4D-consistent data distribution에 적응시키도록 학습
  - Photometric/Geometric Loss
    - 렌더링 Image/Depth Loss로, 복원 품질과 Depth를 통한 3D구조 학습
  - Motion Loss
    - Gaussian Deformation이 3D Point Track의 이동을 따르도록 학습하여 Dynamic Deformation을 학습



< Overview of Diff4Splat >

$$\mathcal{L} = \mathcal{L}_{FM} + \lambda_{photo}\mathcal{L}_{photo} + \lambda_{geo}\mathcal{L}_{geo} + \lambda_{motion}\mathcal{L}_{motion}$$

< Diff4Splat Loss >

Diff4Splat

- Experiments
  - Quantitative Result
    - 30초의 Feed-forward inference로 Optimization-based 방식과 경쟁력 있는 영상 품질 및 4D representation을 생성

Method	Video Generation & Aesthetic Quality					Rec. Time (lower is better)
	FVD ↓	KVD ↓	CLIP-Score ↑	CLIP-Aesthetic ↑	QA-Quality ↑	
Camera-Controlled Video Generation						
CameraCtrl [19]	478.192	8.105	19.365	2.965	1.894	20s
AC3D [2]	339.431	6.342	20.673	3.324	2.158	28s
Explicit 3DGS Representation (Optimization-based)						
AC3D + Shape of Motion† [84]	373.045	6.511	16.201	3.043	1.838	18min
AC3D + SaV† [77]	327.122	5.816	19.018	4.371	2.382	35min
AC3D + Mosca† [32]	235.961	**2.012**	20.214	4.999	**2.842**	45min
Ours (Feed-forward)	**210.153**	2.316	**23.123**	**5.231**	2.813	**30s**

< Quantitative comparison of Diff4Splat >

FVD / KVD: 생성 비디오와 실제 비디오의 Feature Distribution 차이

CLIP Score: Text와의 Semantic Alignment

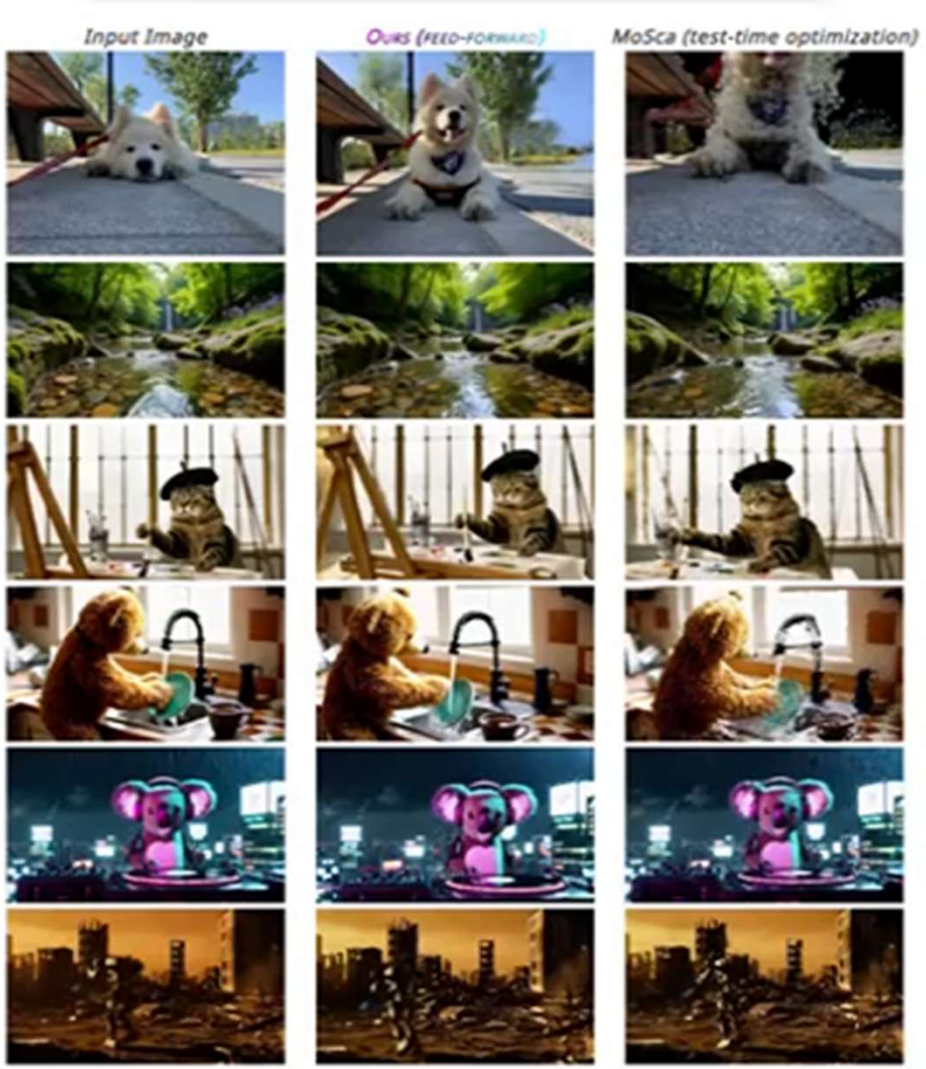
CLIP-Aesthetic: CLIP feature 기반 영상의 미적 품질 예측 점수

QA-Quality: Visual Quality 및 Logical Consistency

RPE Camera Relative Pose 오차

Diff4Splat

- Experiments
  - Qualitive Result



< Video Result >

- 마치며
  - 최근 Video Generation Model이 학습한 appearance / geometry / motion 등의 prior를 영상 생성 이외의 task에 활용하려는 연구가 등장
    - SparseCam4D처럼 생성된 영상을 외재적인 observation / prior로 활용하거나, Diff4Splat처럼 Video Diffusion의 내부 latent representation을 직접 활용하는 등 다양한 접근이 가능
    - 각 task의 목적에 맞게 추가적인 representation, supervision, refinement를 결합
  - 앞으로 Diffusion Model이 학습한 prior를 어떤 형태로 활용하고, 각 task에 맞게 어떻게 refine할 것인가가 새로운 연구 방향이 될 수 있음