Rendering with Diffusion

NeuralField-LDM(CVPR 2023) - Nvidia

Scene Generation with Hierarchical Latent Diffusion Models

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

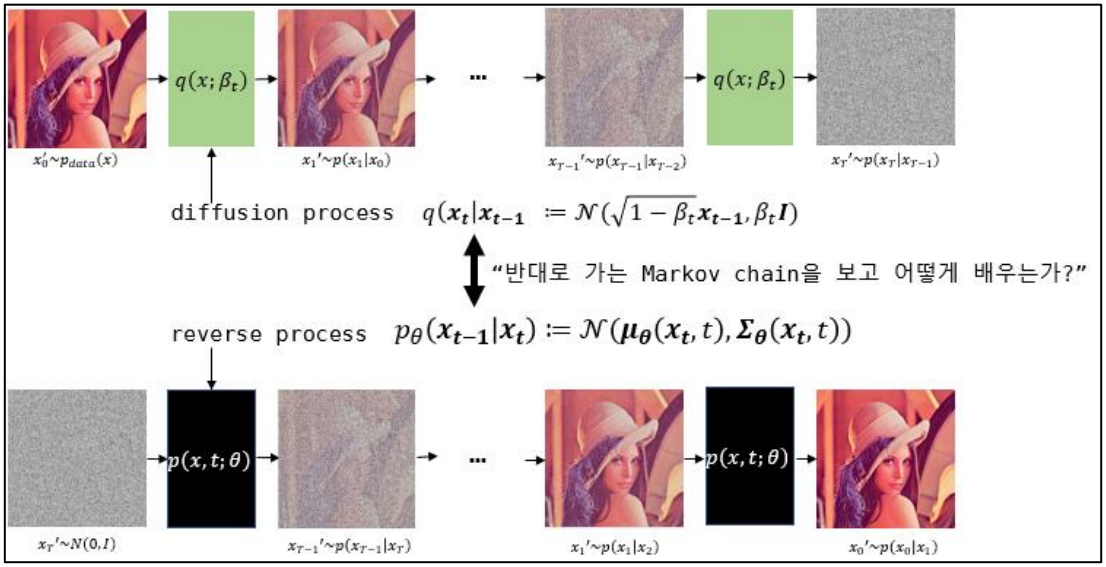
Presented By
신은호

Contents

Introduction

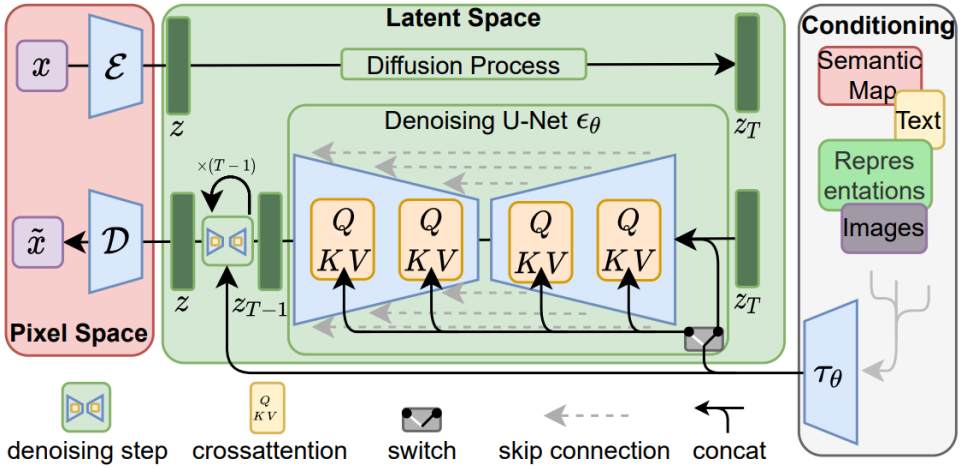
Introduction

- Diffusion Process
  - Concepts
    - Diffusion process
      - Data로부터 noise를 더해가면서 완전한 noise로 만드는 forward process
    - Reverse process
      - Noise로부터 조금씩 복원하며 data를 만들어내는 reverse process 활용
  - Denoising Diffusion probabilistic Models(DDPM)



Introduction

- Latent Diffusion Model
  - Computational efficiency
    - Pixel 공간에서의 최적화 방식은 computational cost efficiency 낮음
    - Latent space에서 diffusion process 응용
  - Latent Diffusion model
    - Real-world를 capture하기에는 표현 차원이 너무나 크기 때문에, Latent이용
    - Stable Diffusion
    - Cross-attention layer 구조 conditioning variables

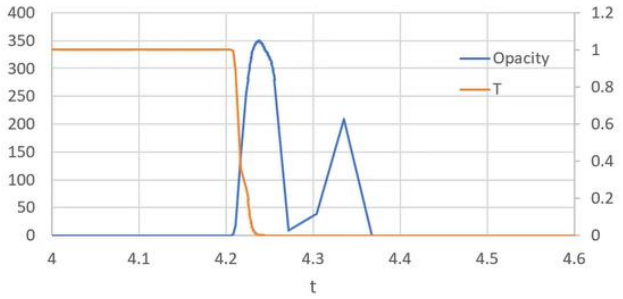


Introduction

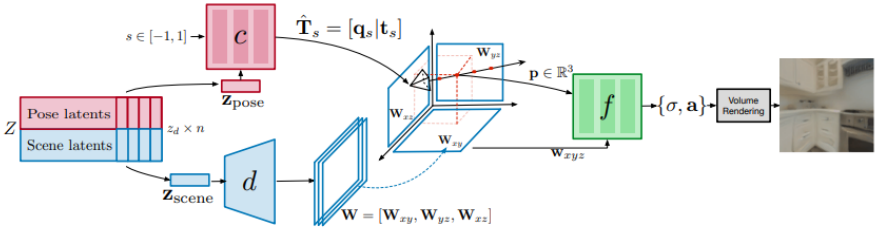
- 3D-Aware Generative Models
  - Camera pose, 객체 모양, 외관 등에 대해 explicit 하게 조작 제공
  - NeRF 기반 Volume Rendering 사용
    - 카메라 position 으로부터 camera ray에 대하여 pixel의 color C(r) 추정

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt$$

$$\text{where} \quad T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

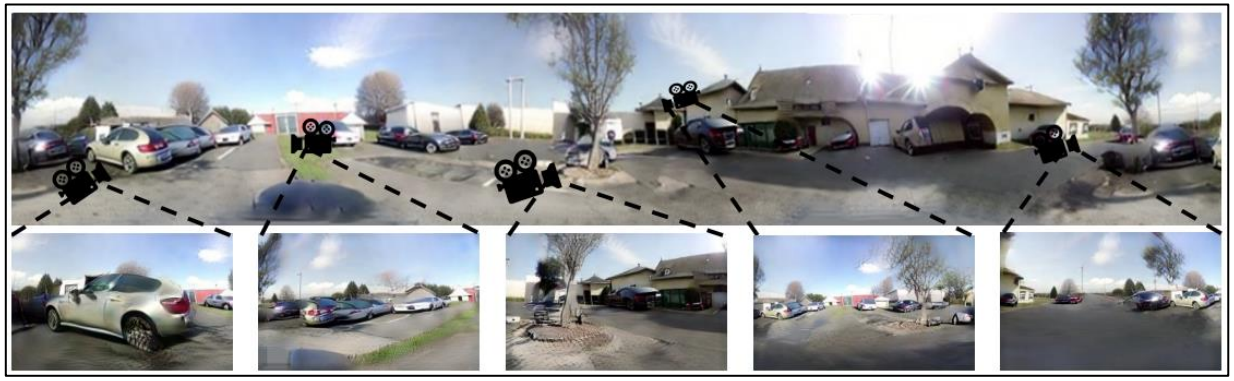


  - GAUDI : A Neural Architect for Immersive 3D Scene Generation
    - 장면을 latent로 변환
    - Latent에 대한 DDM 훈련
    - Auto-Decoder를 통한 scene 생성



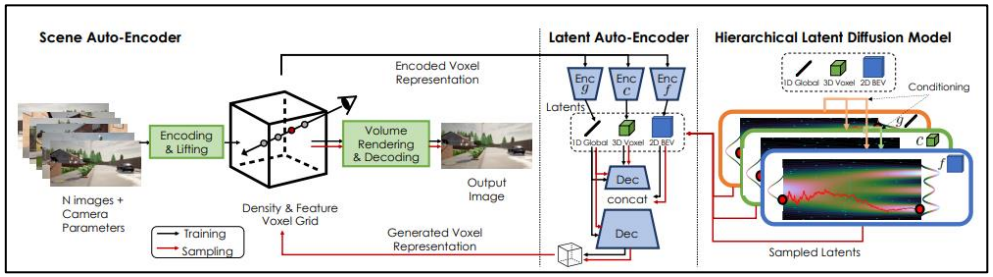
Introduction

- NeuralField-LDM[CVPR 23] (Scene Generation with Hierarchical Latent Diffusion Models)
  - ‘복잡한 open-world 3D 장면을 위한 generative model(생성 모델)’



전역적인 정보를 가진
파노라마 장면 생성 가능

서로 다른 위치에서
촬영된 이미지



NeuralField-LDM Pipeline

1. 이미지, 사전 학습 통해 취득 깊이 정보 Input
2. 2D CNN을 통한 Density & Feature Voxel Grid
3. 1D global latent, 3D coarse latent, 2D fine latent 생성
4. Latent Diffusion model을 활용, latent 학습
5. Voxel 재 생성

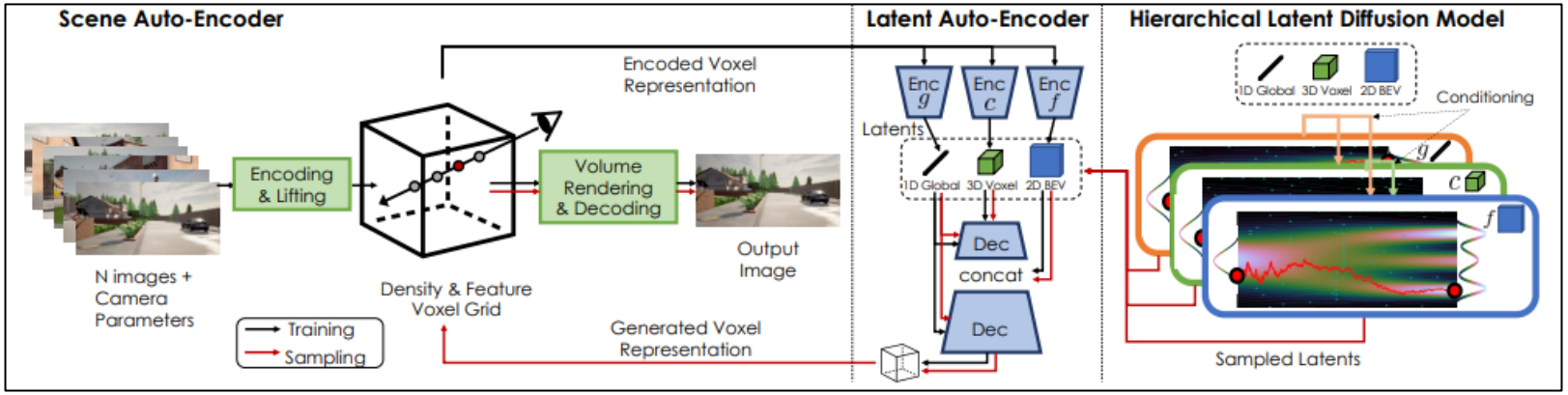
- 주요 내용

  -NeRF’s Volume Rendering & 계층적 Latent Diffusion Model

Main Method

Main Method

- NeuralField-LDM[CVPR 23] (Scene Generation with Hierarchical Latent Diffusion Models)
  - Method 개요



1. Scene Auto-Encoder

- 2D CNN Encoder 이용 feature & density 취득

- Density와 Depth 이용, Occupancy weight 계산

- Occupancy weight 이용, Feature scaling

- Frustums to world coordinates, 공유된 3D 신경장으로 퓨전

2. Latent Auto-Encoder

Encoder
- Voxel을 3개의 Latents로 encode
  - 1D Global g : 전역적인 정보 ex) 시간적 정보
  - 3D Voxel c : coarse 3D 장면 구조
  - 2D BEV(Bird's eye view) f : fine 2D 3D에서 누락된 residual information

Decoder
- Conditional group normalization layer
- g : conditioning variable
- f : intermediate tensor

3. Hierarchical Latent Diffusion Model

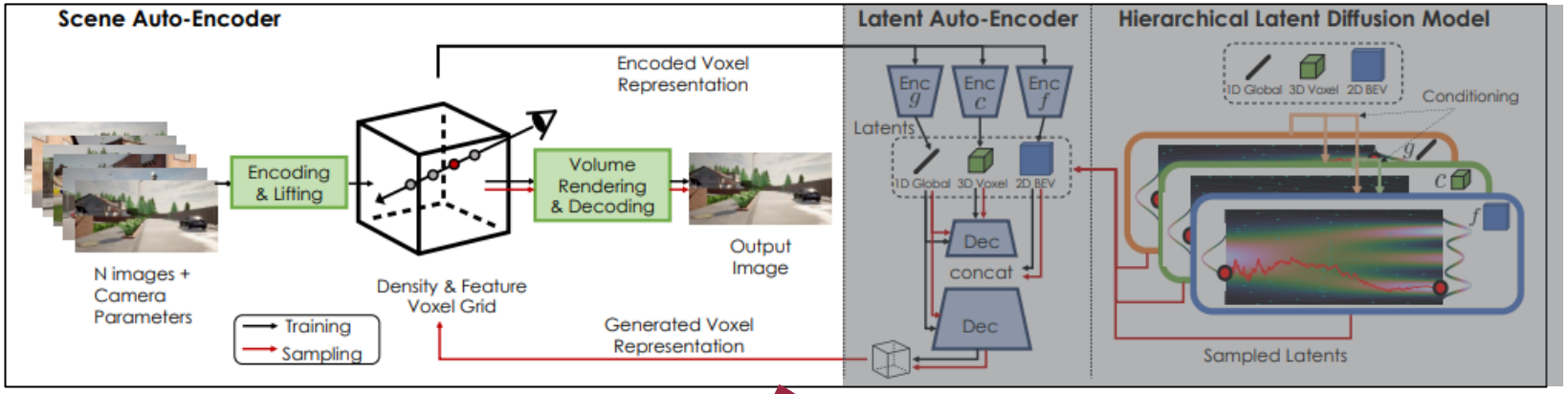
- Generative Adversarial network(GAN) => '훈련 불안정성'

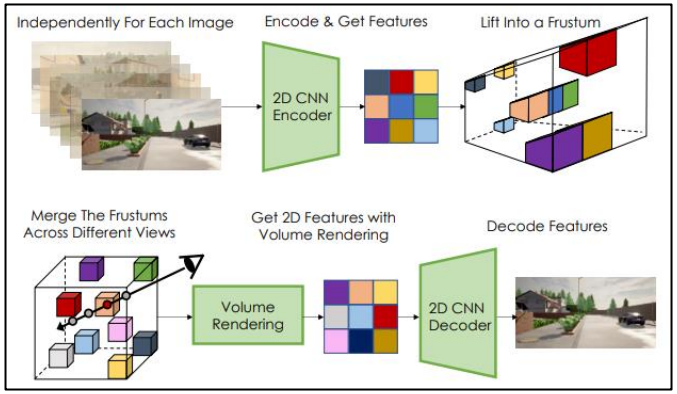
- Denoising Diffusion models(DDMs) => 대규모 데이터셋이 필요, 제한된 시나리오

- (결론) Latent Diffusion Model => 중간 단계 latent 분포 생성 후 diffusion model의 latent distribution fitting

Main Method

- Scene Auto-Encoder





Main Method

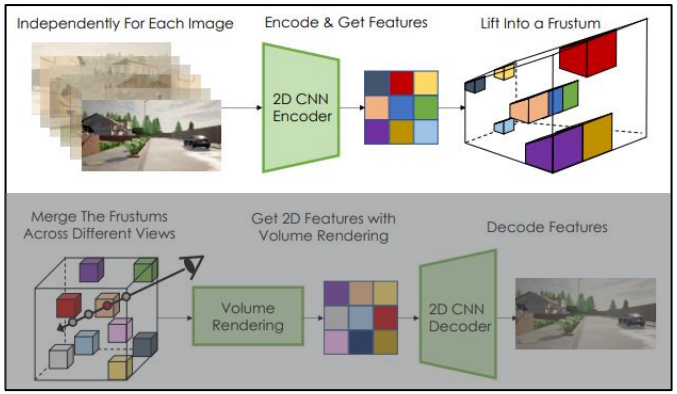
- Scene Auto-Encoder
  - 2D CNN Encoder, H x W x (D + C) 차원 획득
    - H, W는 이미지의 height width보다 작게 설정
    - D : density value
    - C : feature vector
  - 카메라 포즈 k에 따른 H x W x D size frustum (D : Depth)
    - Density value σ, Distance between each depth δ 이용, Occupancy 계산

$$O(h,w,d) = \exp(-\sum_{j=0}^{d-1} \sigma_{(h,w,j)}\delta_j)(1-\exp(-\sigma_{(h,w,d)}\delta_d))$$

    - Occupancy로 scaling되어 Feature ϕ(h, w) Frustum 저장

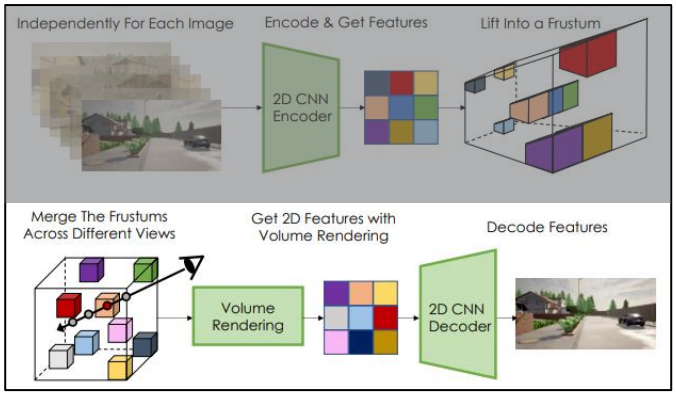
$$F(h,w,d) = [O(h,w,d)\phi(h,w), \sigma(h,w,d)]$$

    - 이후, shared 3D neural field(x, y, z) $V_{Density}$, V_{Feat}로 fuse
      - Frustums to world coordinates



Main Method

- Scene Auto-Encoder
  - Volume rendering
    - k, V 이용 2D feature map projection
      - k : 카메라 pose
      - Feature, density trilinearly interpolate
    - 2D feature를 CNN decoder에 fed
    - Rendering 된 image 획득 가능
      - $\hat{\imath} = r(V, k)$
    - Expected depth $\hat{\rho}$ 추정 가능
  - Loss
    - Image Reconstruction loss $||i - \hat{\imath}||$
      - Perceptual Loss (LPIPS)
      - Better than L1 or L2 loss
    - Depth supervision loss $||\rho - \hat{\rho}||$
      - MSE Loss



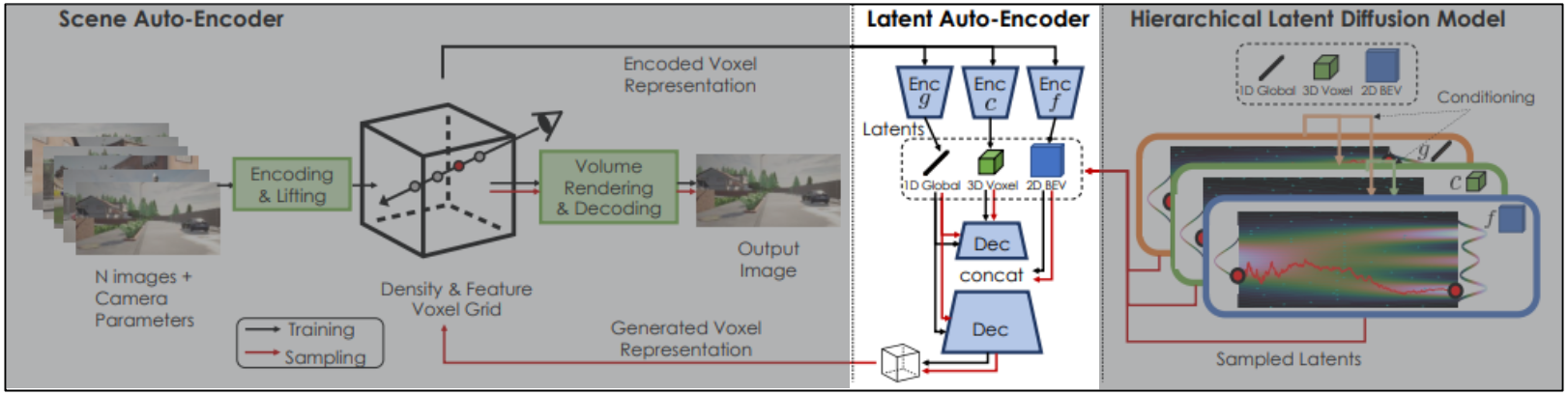
LPIPS

비교할 2개의 이미지를 각각 VGG Network에 넣고, 중간 layer의 feature값들을 각각 뽑아내서, 2개의 feature가 유사한지를 측정하여 평가지표로 사용

(The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition,)

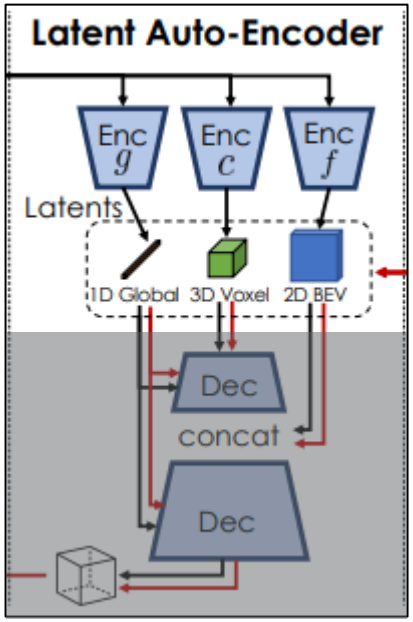
Main Method

- Latent Voxel Auto-Encoder

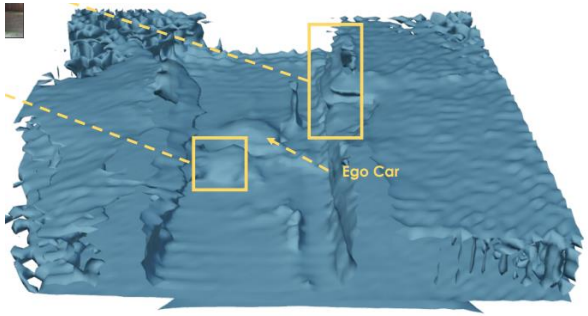


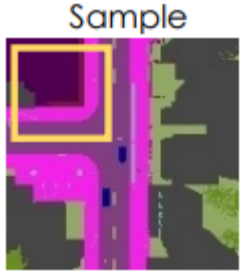
Main Method

- Latent Voxel Auto-Encoder
  - Channel 따라 $V_{density}$, V_{feat} 결합
  - Encode voxel into three latents with 2D CNNs
    - 1D global latent g
      - Global properties(the time of the day)
    - 3D coarse latent c
      - 3D scene structure
    - 2D fine latent f
      - Same horizontal size X x Y as V
      - Bird's eye view



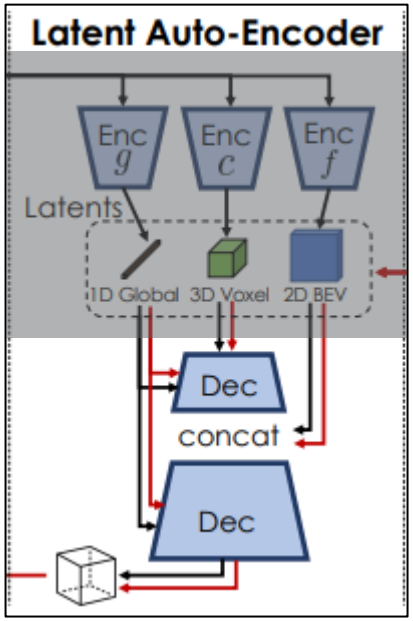
- 구현 detail
  - 3D CNN 대신 2D CNN 구조 이용
    - V's vertical axis channel dimension
  - Regularization
    - g : Small KL-penalty
    - c : Codebook 1024 entries
    - f : Codebook 128 entries





Main Method

- Latent Voxel Auto-Encoder
  - CNN decoder
    - Latent c : vertical axis에 따라 concatenate
    - Conditional group normalization layers with g
    - Concatenate f intermediate tensor
    - Latent decoder output $\hat{V}$ reconstructed voxel
  - Loss
    - Voxel reconstruction loss
      - $\|V - \hat{V}\|$
        - ✓Density Voxel 2.5배 higher weight
        - ✓To reconstruct the geometry
    - Image reconstruction loss
      - $\big|\|i - \hat{i}\|\big|, \hat{i} = r(\hat{V}, k)$
        - ✓LAE의 학습에만 사용
        - ✓Scene Auto-Encoder is kept fixed

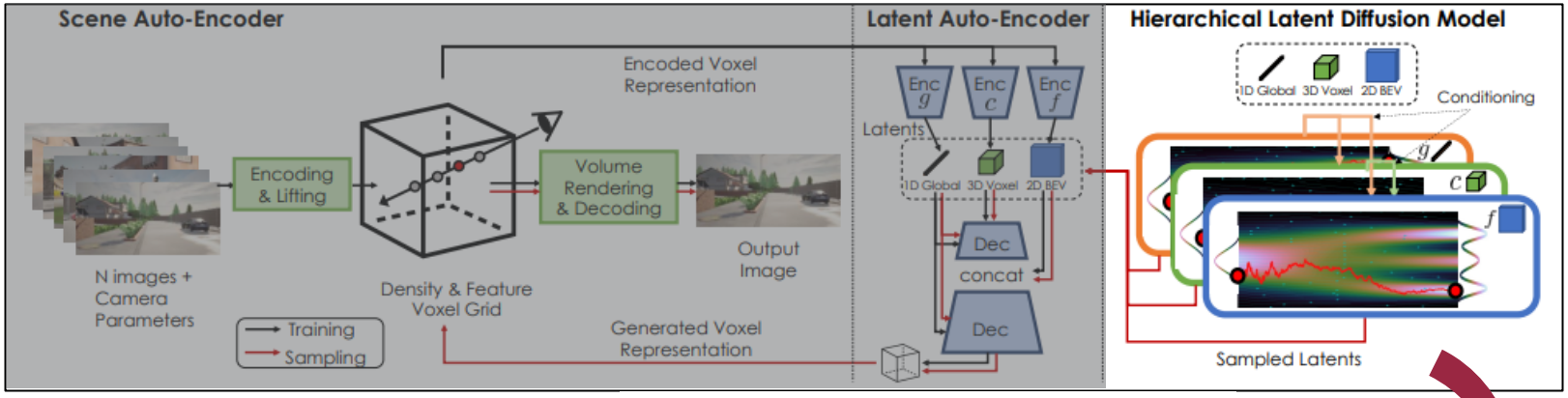


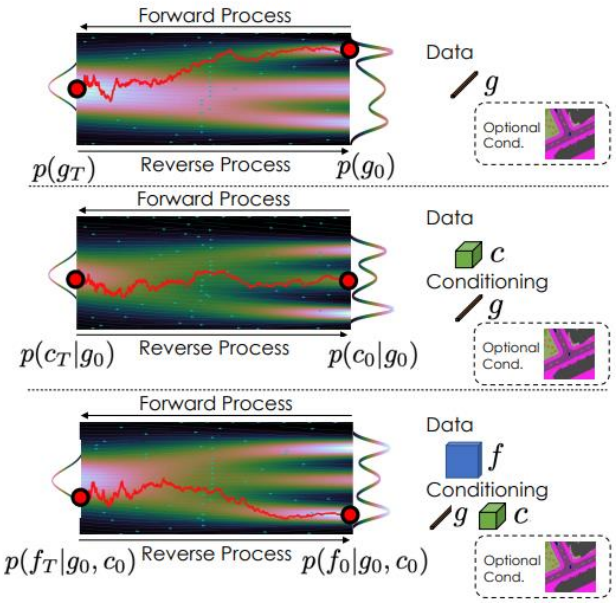
Layer	Output dimension
Input c (3D)	$4 \times 8 \times 32 \times 32$
Concat Z-axis	$(4 \times 8) \times 32 \times 32$
Conv2D 3×3	$512 \times 32 \times 32$
MidBlock-CGN	$512 \times 32 \times 32$
ResBlock-CGN	$512 \times 32 \times 32$
ResBlock-CGN	$512 \times 32 \times 32$
ResBlock-CGN	$512 \times 32 \times 32$
2 × {ResBlock-CGN	
ResBlock-CGN	
ResBlock-CGN	$512 \times 128 \times 128$
Upsample2×}	
Combine f	$512 \times 128 \times 128$
Conv2D 3×3	$1024 \times 128 \times 128$
Split Z-axis (3D)	$32 \times 32 \times 128 \times 128$

Table 8. Decoder of the latent auto-encoder

Main Method

- Hierarchical Latent Diffusion Model



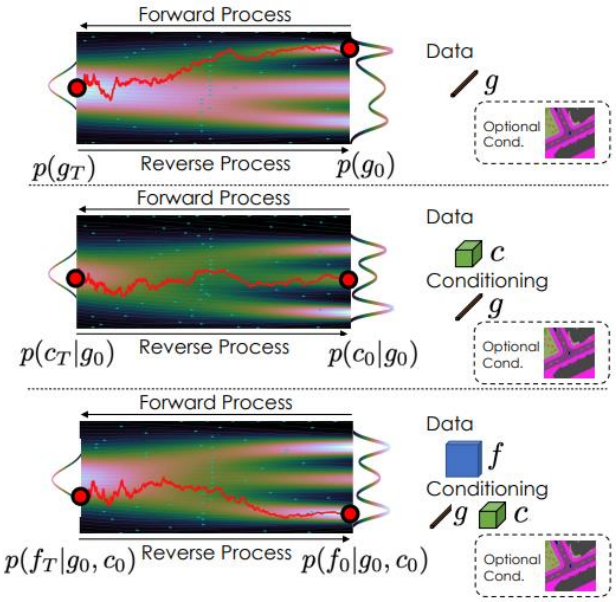


Main Method

- Hierarchical Latent Diffusion Model
  - Denoising Diffusion Models(DDMs)
    - $p(V, g, c, f) = p(V|g, c, f)p(f|g, c)p(c|g)p(g)$
      - 계층적 구조를 통해 Voxel V 추정 가능
    - $\mathbb{E}_{t,\epsilon,g_0}\left[w(\lambda_t)||g_0 - \psi_g(g_t, t)||_2^2\right]$

      $\mathbb{E}_{t,\epsilon,g_0,c_0}\left[w(\lambda_t)||c_0 - \psi_c(c_t, g_0, t)||_2^2\right]$

      $\mathbb{E}_{t,\epsilon,g_0,c_0,f_0}\left[w(\lambda_t)||f_0 - \psi_f(f_t, g_0, c_0, t)||_2^2\right]$
    - Ψ는 학습 가능한 denoising network
    - Latent Voxel Auto-Encoder
      - 한 번 g, c, f가 sample 후, latent decoder 이용 voxel V 생성 가능
    - Scene Auto-Encoder
      - Volume rendering과 decoding step 이후, desired viewpoints scene 생성

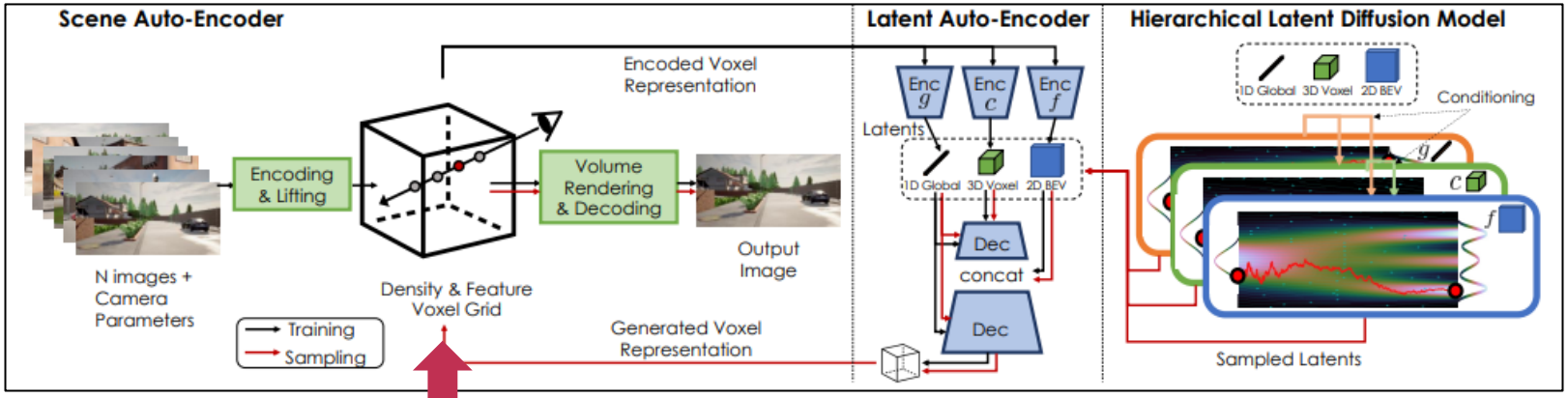


Main Method

- Hierarchical Latent Diffusion Model
  - 구현 세부사항
    - 1D global latent g를 위한 ψ는 skip connection을 가진 linear layer
      - Global latent g는 c와 f를 위한 ψ에 conditional group normalization layer로 제공
      - trajectory 정보는 모델이 ‘where to focus on generating’에 유용한 정보
        - ✓trajectory information이 g에 concatenate
    - 2D, 3D latent c, f는 U-net 구조
    - Ψ는 conditioning variable을 cross-attention layers로 input
      - Self-attention layer 혹은 extra conditioning information(BEV map)
    - 각 ψ는 한 번 학습된 이후에는 parallel하게 학습 가능

Main Method

- Post-Optimizing Generated Neural Fields



Stable Diffusion

$$\nabla_V L_{SDS} = \mathop{\mathbb{E}}_{\epsilon,t,\kappa}\left[w(\lambda_t)(\epsilon - \hat{\epsilon}_\theta(r(V,\kappa),t))\frac{\partial r(V,\kappa)}{\partial V}\right]$$

Main Method

- Post-Optimizing Generated Neural Fields
  - Real-world data를 통해 reasonable texture와 geometry 획득 가능
    - But, 대량 데이터 통해 학습된 2D image diffusion model을 이용, 추가 optimizing 가능

Main Method

- Post-Optimizing Generated Neural Fields
  - SDS(Score Distillation Sampling) Loss
    - 초기 voxel V에서 반복적으로 update
    - Scene으로 부터 viewpoint rendering하며 각 이미지에 대해 SDS Loss 이용
    - $6m^2$에서 장면의 origin 주위로 uniformly sample

$$\nabla_V L_{SDS} = \mathop{\mathbb{E}}_{\epsilon,t,\kappa}\left[w(\lambda_t)(\epsilon - \hat{\epsilon}_\theta(r(V,\kappa),t))\frac{\partial r(V,\kappa)}{\partial V}\right]$$

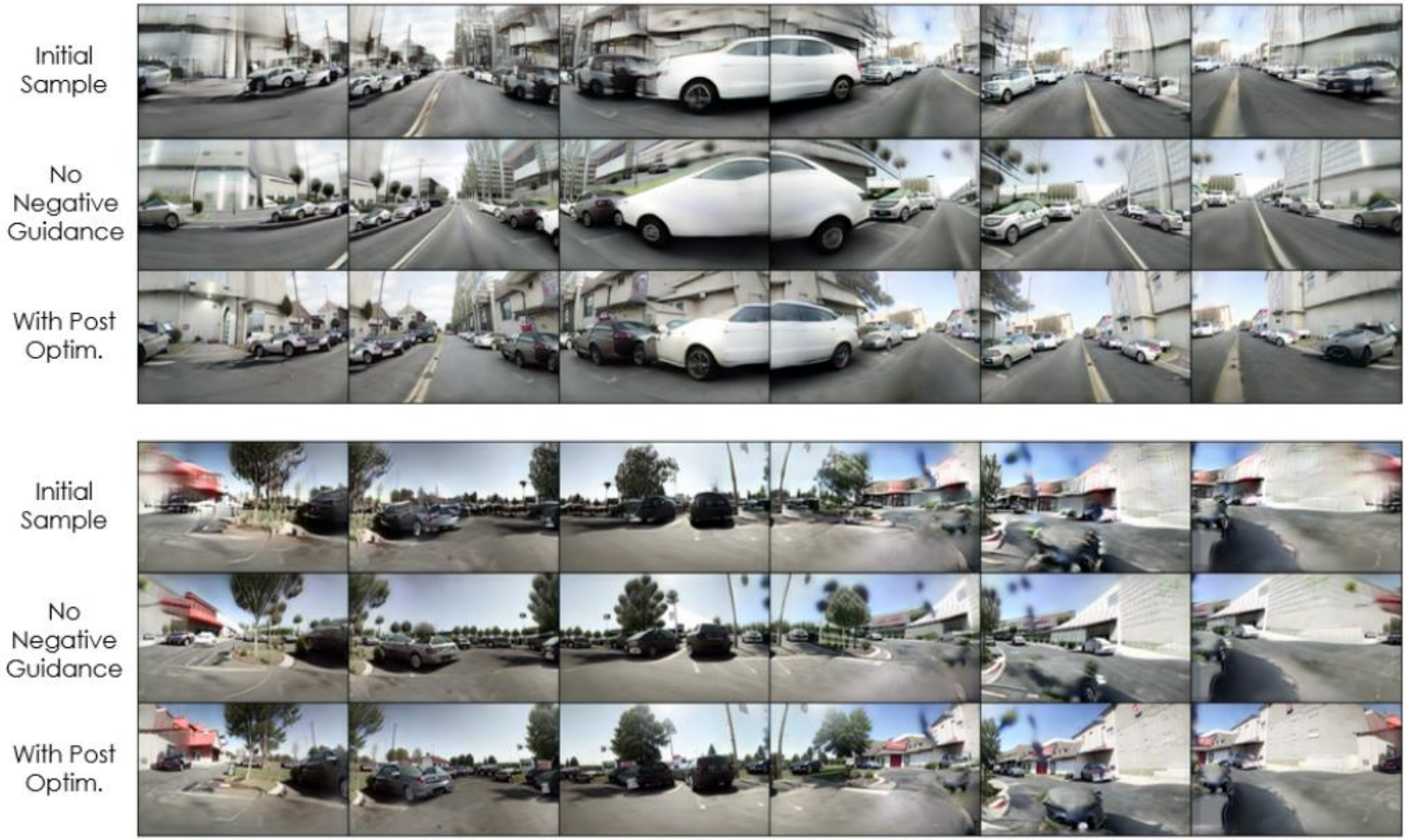
    - $\hat{\epsilon}_\theta$는 off-the-shelf latent diffusion model
    - CLIP image embeddings를 통해 finetuned
      - CLIP이 이미지의 품질에 대한 표현을 포함하고 있음을 발견
      - Initial rendering image의 Geometry distortion과 texture errors가 유사하게 발생
      - Negative guidance 이용 SDS Loss optimizing

Main Method

- Post-Optimizing Generated Neural Fields
  - SDS Loss with negative guidance
    - Dataset image로 부터 얻은 clean image conditioning CLIP embeddings : y
    - Samples로부터 얻은 artifact conditioning CLIP embeddings : y'
    - Classifier-free guidance
      - Conditioning vector y, Unconditional embedding with y'
    - $\frac{p(x|y)^{\alpha}}{p(x|y`)}$ 에서 샘플링 하는 것과 동일 $\nabla_x \log \frac{p(x|y)^{\alpha}}{p(x|y')} = \alpha \nabla_x \log p(x|y) - \nabla_x \log p(x|y')$

Main Method

- Post-Optimizing Generated Neural Fields
  - Result



Result

Result

- Baseline Comparisons
  - FID(Frechet Inception Distance)
    - GAN으로 생성된 output과 real data 비교 지표
    - Pretrained inception Network의 conv layer 통과 activation 간 비교
  - FVD(Frechet Video Distance)
    - 두 비디오 시퀀스 간 차이 측정
    - FID의 확장판
    - 프레임 간 시간적 상관관계 고려
- 참고
  - EG3D and GSN (outdoor)
    - GAN-based 3d generative model
  - GSN(FVD for indoor)
    - Inconsistent rendering

Criterion	Method	Depth	VizDoom	Replica
FID (↓)	GRAF [66]	✗	47.50	65.37
	π-GAN [5]	✗	143.55	166.55
	GSN [9]	✓	37.21	41.75
	GAUDI [3]	✓	33.70	18.75
	NF-LDM	✓	**19.54***	**14.59**

Table 1. FID [22] scores on VizDoom and Replica. NF-LDM outperforms all baseline models. Baseline numbers are from [3].

Criterion	Method	Depth	Carla	AVD
FID (↓)	EG3D [6]	✗	76.89	194.34
	GSN [9]	✓	75.45	166.07
	NF-LDM	✓	**35.69**	**54.26**

Table 2. FID [22] scores on Carla and AVD datasets. Baseline models have trouble learning the distribution of complex outdoor datasets, in particular AVD, while NF-LDM models them well.

Criterion	Method	Depth	Carla	AVD
FVD (↓)	EG3D [6]	✗	134.94	1232.38
	GSN [9]	✓	184.30	1659.81
	NF-LDM	✓	**91.80**	**242.50**

Table 3. FVD [77] scores on Carla and AVD Datasets. As for FID, baseline models have trouble learning to model complex datasets.

Result

- Baseline Comparisons
  - Outdoor Scene

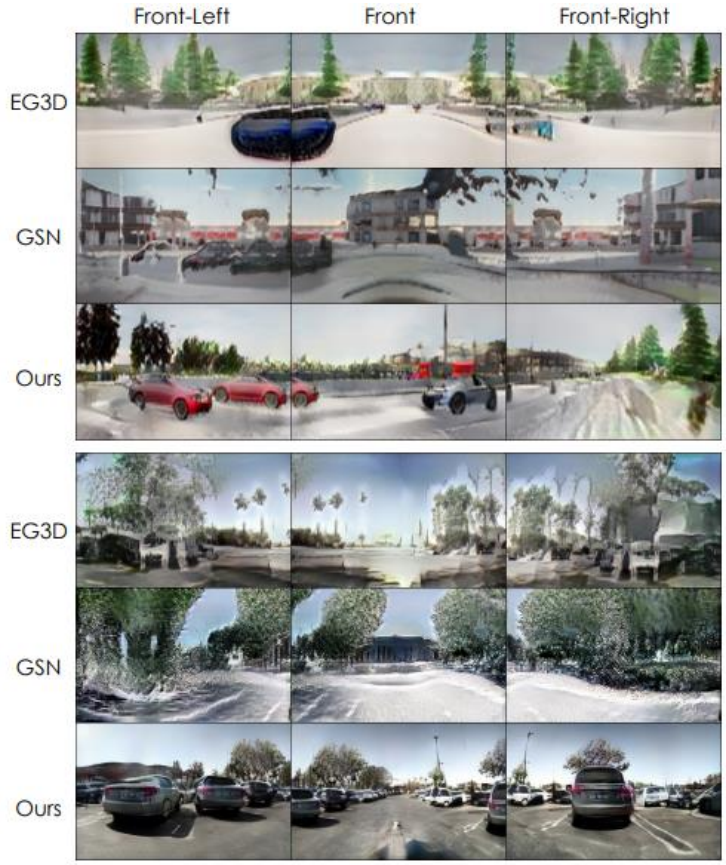


Figure 6. **Generated Scenes:** The top three rows are samples from Carla, and the bottom three rows are samples from AVD.

  - Indoor Scene

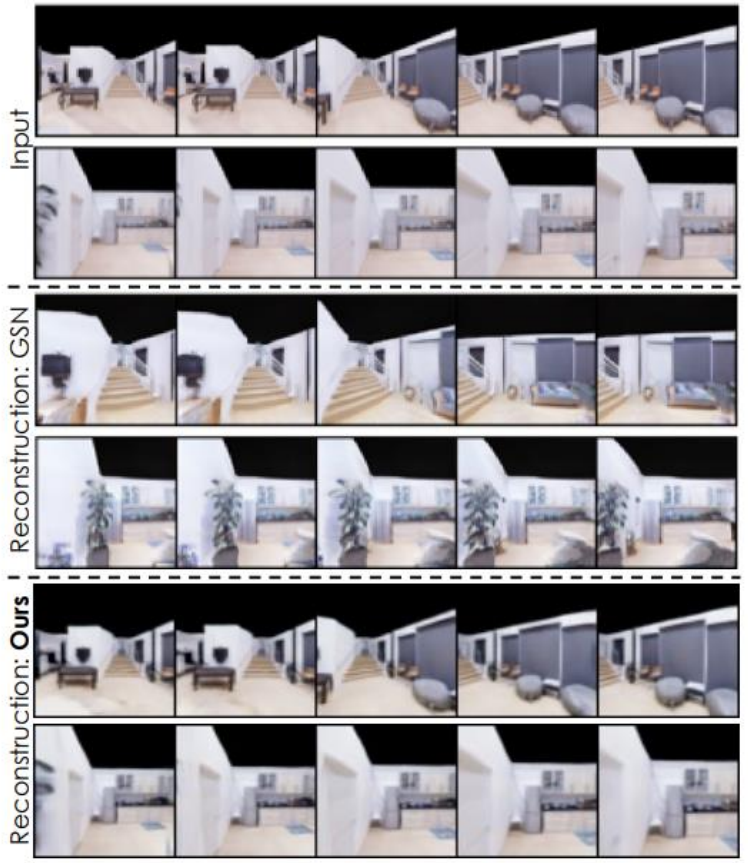


Figure 8. Reconstructing held-out scenes not seen during training.

Result

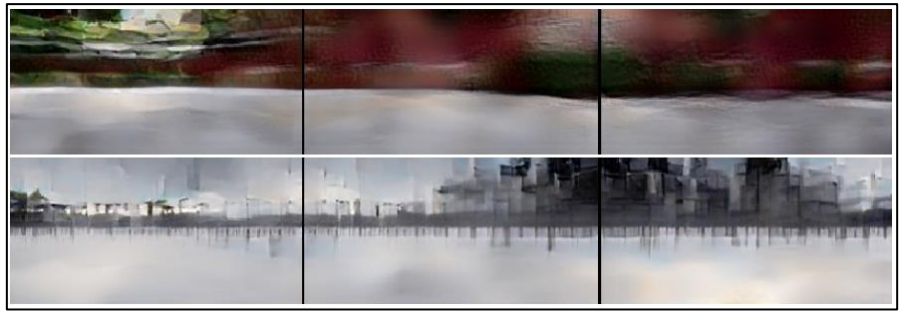
- Ablations
  - Hierarchical structure
    -

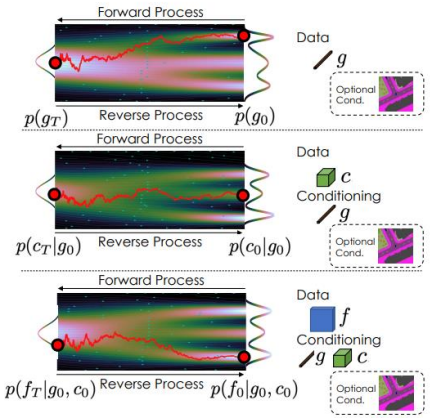
	Coarse lat. c	+ Fine lat. f	+ Global lat. g
FID (↓)	46.43	43.52	**35.69**

      - Global latent g가 FID 지표에 큰 영향을 주는 것을 확인 가능
  - Scene encoder의 voxel size
    - Voxel size가 scene의 detail을 포착하는데 중요할 것으로 판단
      - Voxel과 encoder frustum의 size가 커질수록 pixel-level detail 증가 기대
    -

	$32 \times 32 \times 8$	$64 \times 64 \times 16$	$128 \times 128 \times 32$
Percept. Loss (↓)	0.3508	0.2688	**0.2237**

      - Generative model에게 있어서 high-dimensional voxel modeling은 challenging
      - 128 x 128 x 32가 안정적인 품질을 만드는 가능한 작은 크기임을 확인 가능



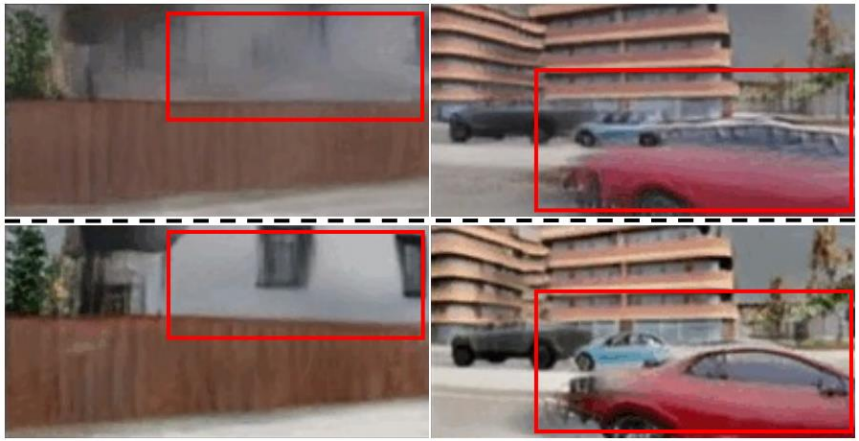


Result

- Ablations
  - Latent Encoder
    - 더 큰 voxel은 더 나은 reconstruction을 제공하지만, 생성 모델의 부담 증가
    - Latent encoder의 downsampling 크기 비교
    -

	$ds = 16$	$ds = 8$	$ds = 4$
Vox. Recon Loss (↓)	0.6076	0.5949	**0.4915**

    - Downsampling factor(for coarse 3D latent)가 4일 때, 충분히 표현 가능한 범위 임을 확인
  - Explicit Density
    - Occupancy 명시적 예측을 통해, frustum 병합의 부정확한 특징 혼합 방지
    - 각 시점에서 밀도를 예측함으로써 추가적인 밀도 예측 단계 회피



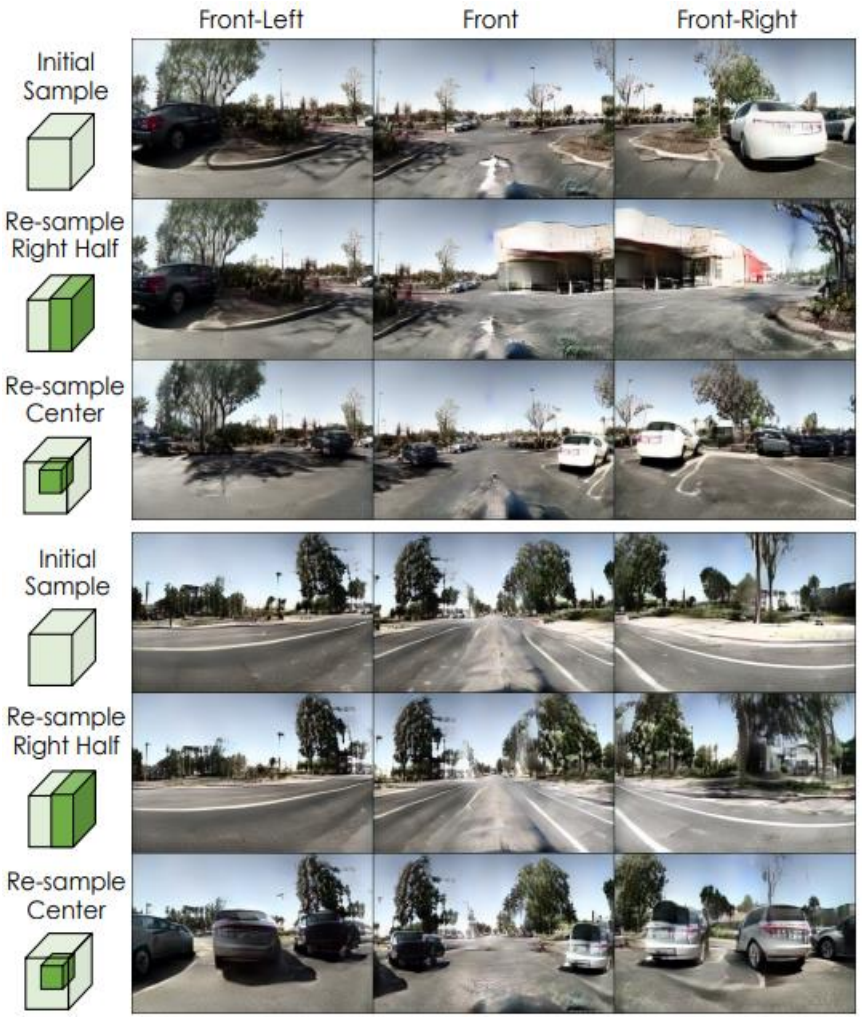
Conclusion

Conclusion

- Applications
  - BEV Conditional Synthesis
    - Additional conditioning signal 이용
      - BEV segmentation map
    - Cross attention layer 이용
    - BEV map conditioning에 따라 reconstruction 결과 변경 확인 가능

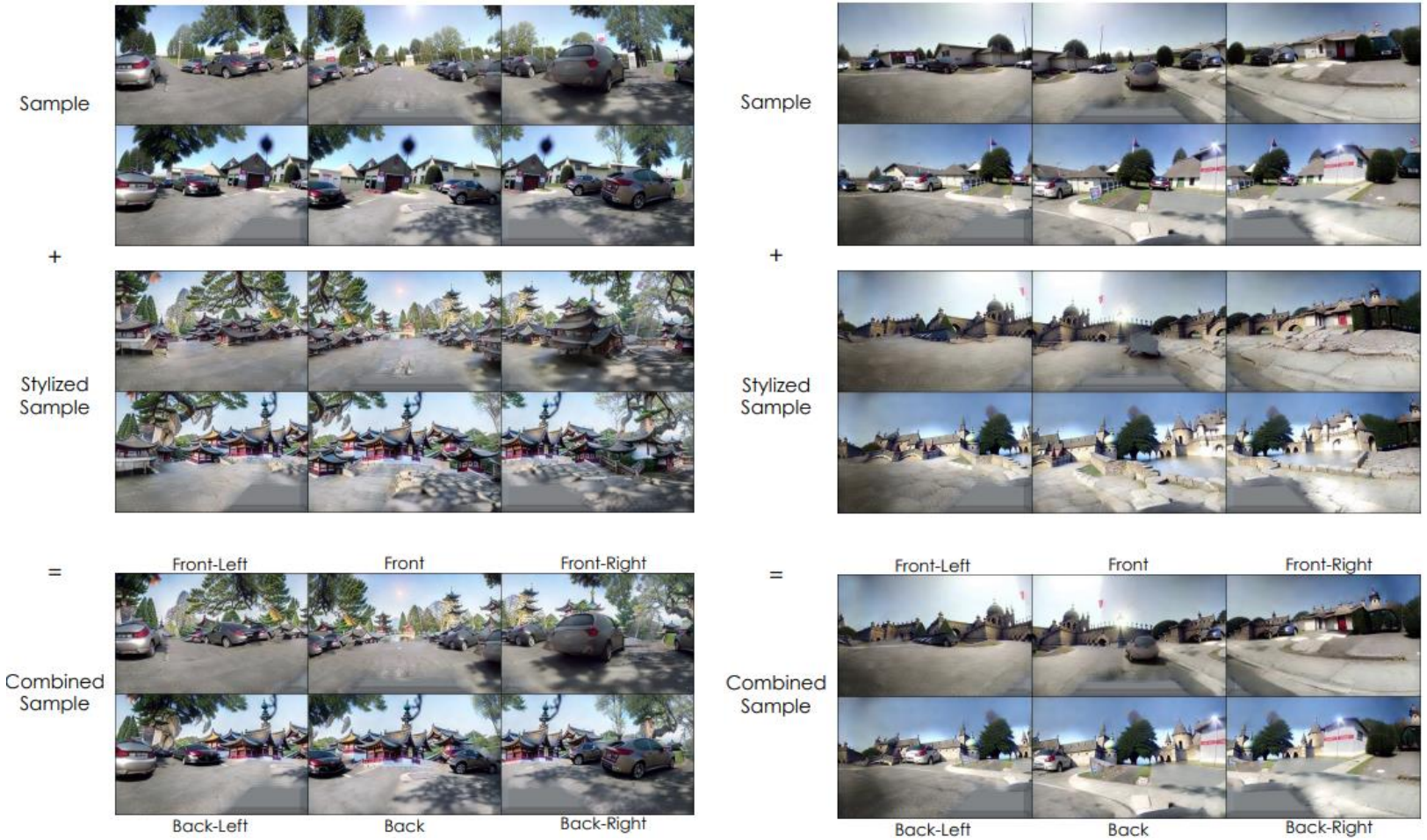
Conclusion

- Applications
  - Scene Editing in 3D coarse latent
    - Image inpainting without explicit training
    - 3D coarse latent c의 특정 영역 resampling
      - 각 denoising step에서 보존할 영역에 noise 추가, sampling 중인 영역과 연결하여 diffusion model 통과
    - 새로운 c를 얻은 후, fine latent 또한 conditioned on c로 re-sample



Conclusion

- Applications
  - Stylization(Post-Optimization)



Conclusion

- Limitations
  - NeuralField-LDM의 구조는 고품질 생성 및 재구성을 가능
    - 학습 시간과 sampling 속도의 감소 발생
  - Neural field의 voxel grid 표현으로 volume render와 diffusion model 비용 증가
    - 대안 sparse representations 필요
    - More efficient sparse voxel representations
      - 대규모 실세계 데이터
      - 지속적으로 증가 가능한 형태
  - Multiview 이미지 요구
    - Universal problems와 overfitting 문제 위험 존재
    - AVD의 출력 sample의 다양성이 제한되었던 것은 limited number of scenes 때문

감사합니다.

Dataset

- VizDoom
  - Front view sensor observations obtained by navigating inside a synthetic game environment
- Replica
  - Reconstructions of 18 indoor scenes, containing 101 front-view trajectories
- Carla
  - Open-source simulation platform for self-driving research
  - Six cameras (front-left, front, front-right, back-left, back, back-right), covering 360 degrees with some overlaps
- AVD
  - In-house dataset of human driving recordings in roads and parking lots.

Reference

- https://jang-inspiration.com/ddpm-2

- https://dlaiml.tistory.com/entry/Score-based-Generative-Models%EA%B3%BC-Diffusion-Probabilistic-Models%EA%B3%BC%EC%9D%98-%EA%B4%80%EA%B3%84

- https://xoft.tistory.com/4