High-Resolution Image Synthesis With Latent Diffusion Models

2023-03-21

김미주

INTRODUCTION

- Image Synthesis 연구
  - Image Synthesis 는 최근 가장 많은 발전을 이룬 computer vision 분야 중 하나이며, 가장 큰 computational 수요를 가진 분야
- 기존 연구의 한계점
  - (GANs)
    - 복잡한 multi-modal distributions 으로 scale 하기 힘든 adversarial learning 으로 인해 제한된 variability 를 가진 data 에 국한
  - DMs
    - pixel space 에서 작동하기 때문에 powerful DMs 의 optimization 은 많은 GPU 가 필요하고 sequential evaluations 로 인해 긴 inference 시간이 필요

INTRODUCTION

Departure to Latent Space

- 대략적인 학습 단계(two stages)
  a. **perceptual compression** stage
     - high-frequency details 을 제거하지만 아직 semantic variation 은 학습하지 않음.
  b. **semantic compression** stage
     - 실제 generative model 이 데이터의 semantic and conceptual composition 학습함.

- perceptual and semantic compression
  - digital image 의 대부분의 비트 : imperceptible details
    - DM은 responsible loss term 을 최소화하여 이러한 의미 없는 정보(imperceptible details) 제거 가능
      그러나, (during training) gradients 및 neural network backbone (training and inference) 은 여전히
      모든 pixel 에서 평가되어 과도한 computations 이 요구됨.

INTRODUCTION

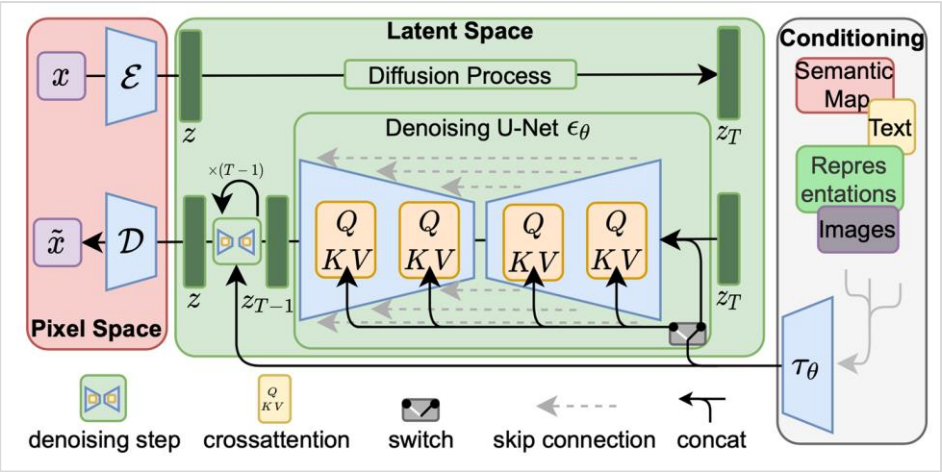
Departure to Latent Space

- 논문의 목표
  - perceptually equivalent + computationally suitable space 찾아서 high-resolution image synthesis 을 위한 diffusion models 훈련
- **Latent Diffusion Models(LDMs)**
  - 💡 Diffusion Model을 pixel space 가 아닌 pretrained autoencoders 의 latent space 에 적용 + cross-attention layers 도입
    → latent diffusion models (LDMs) 제안
  - data space 와 perceptually equivalent 하면서도 lower-dimensional representational space 제공하는 autoencoder 학습
    → 이전 연구들과 다르게 spatial compression 에 의존할 필요가 없다.
    ? : spatial dimensionality(공간적 차원; pixel space) 보다 더 좋은 scaling properties 을 가지고 있는 latent space 에서 DM 을 학습하기 때문에
  - lower dimensionality 인 compressed latent space 작동함으로써 2가지 단점을 해결함.
    → 컴퓨팅 리소스 ⬇ +synthesis quality 를 유지하면서도 inference speed ⬆

Method

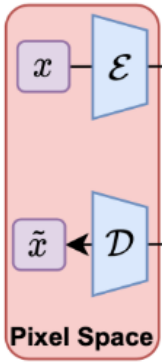
- perceptual compression : autoencoder
  - image space 와 인지적으로(perceptually) 동등한 space를 학습하면서도 computational complexity 가 크게 감소하는 autoencoding model 을 활용
  - 장점
    - high-dimensional image space 에서 벗어나 low-dimensional space 에서 sampling 이 수행되므로 훨씬 효율적인 DMs를 얻음.
    - UNet architecture based DM 의 inductive bias 을 활용하여 spatial structure 를 가진 data 에 특히 효과적이다.
      → 기존 연구에서 요구하는 높은 spatial compression 을 완화

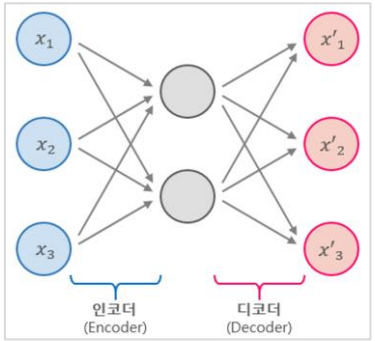
- **LDMs model**



Perceptual Image Compression (Pixel Space ↔ Latent Space)

- **perceptual compression model: AutoEncoder (ε,D)**





- **인코더(encoder) ε**: x → z(latent representation)
- **디코더(decoder)** D: z(latent representation) → x˜

고차원의 이미지(x)를 잘 표현하는 manifold 인 latent representation z 추출
→ reconstruction 이미지(x˜) 로 복원

data

- **x** ∈ $R^{\{H\times W\times 3\}}$: RGB 공간의 이미지
- encoder **ε** : x 를 latent representation z = **ε**(x)으로 인코딩 ⇒ x downsampling (factor f = H/h = W/w)
- **z** ∈ $R^{\{h\times w\times c\}}$ = **ε(x)** : latent representation (2-dimensional structure)
- decoder **D** : latent representation(z) 를 single pass로 decoding 하여 x˜ = D(z) = D(**ε**(x))를 제공
- **x˜** = D(z) = D(E(x)) : reconstruction 이미지

본 논문에서는 각기 다른 downsampling factor f = 2^m, m ∈ N 에 대해 실험

Latent Diffusion Models

- **Diffusion Model**
  - generative model(GAN, VAE, Flow-based models, Diffusion models)
    : 새로운 data instance를 생성해내는 모델 ⇒ Training Data distribution에 근사하는 특성

  - 2가지 가정
    1. 이산 마코프 가정(Discrete Markov Process Assumption)

$$P\left[s_{t+1} \mid s_t\right] = \mathrm{P}\left[s_{t+1} \mid s_1, \ldots, s_t\right]$$

    - Markov 성질 : “특정 상태의 확률(t+1)은 오직 현재(t)의 상태에 의존한다.”
    - 이산 확률과정 : 이산적인 시간(0초, 1초, 2초, ..) 속에서의 확률적 현상

    1. 정규성(Normality) 가정 : “특정 데이터가 정규 분포를 따를것이다.”

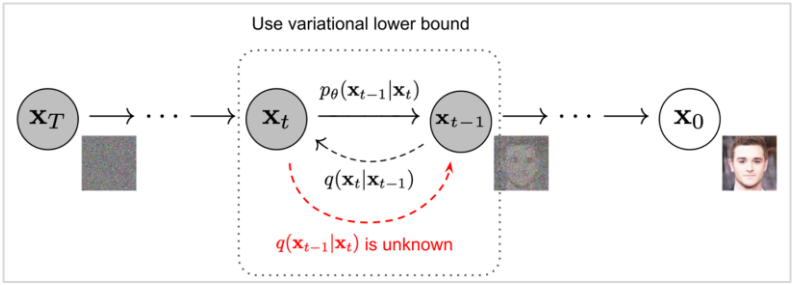
    → 1+2 : 평균과 분산(μ,Σ)이라는 두가지 모수에 의해 결정되는 정규분포 그래프로서 Diffusion Model에서는 각 확률 단계가 Normal Distribution을 따를것이라 가정

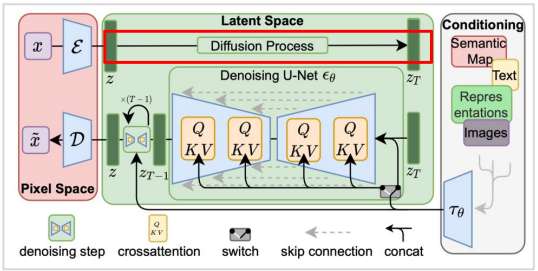
$$p_\theta\left(X_{t-1} \mid X_t\right) = N\left(X_t; \mu_{X_{t-1}}, \Sigma_{X_{t-1}}\right)$$

- 2가지 가정과 Diffusion Model
  Markov 가정과 간단한 분포(정규분포)를 단계별로 활용하여 점차 복잡한 데이터를 표현하는 것이 Diffusion Model의 핵심

Latent Diffusion Models

- **Diffusion Model**
  denoising process 를 통해 data distribution p(x)를 학습하는 probabilistic model
  → denoising process : length T 의 fixed Markov Chain의 reverse process 학습





- **Forward diffusion process q**

  : 샘플 이미지 x_0가 시점 0-T 까지 작은 gaussian noise를 줘서 최종적으로 노이즈로 이루어진 x_T를 만드는 과정

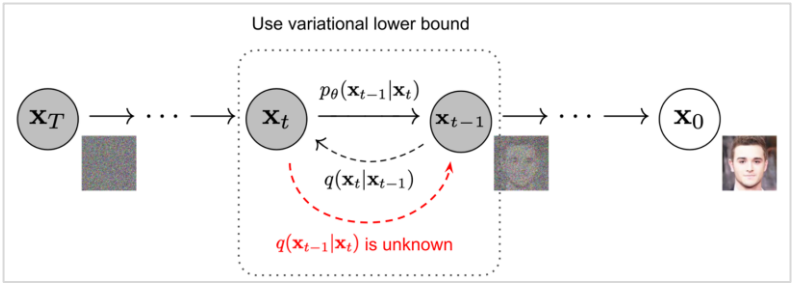
  - noise: 정규성 가정에 따라 정규 분포 형태를 따르는 임의의 Gaussian Noise 주입
  - x_1 : x_0에 noise 적용한 이미지 → $q(x_1|x_0)$
    time t에 대해 general 하게 표현한다면 $q(x_t|x_{t-1})$으로 표현할 수 있다.

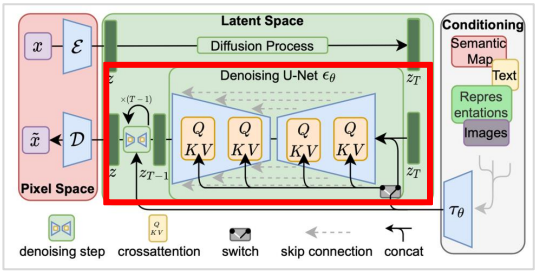
    LDMs: latent representation z = ε(x) (latent space에 매핑) → z_T(noised latent representation)

→ t-1에서 t번째 이미지가 되는 과정은 정의한 노이즈에 따라서 바로 알아낼 수 있어 학습이 필요 없지만 반대 과정은 알 수 없기 때문에 학습이 필요

Latent Diffusion Models

- **Diffusion Model**
  denoising process 를 통해 data distribution p(x)를 학습하는 probabilistic model
  → denoising process : length T 의 fixed Markov Chain의 reverse process 학습





  - **Reverse process**

    : $q(x_t|x_{t-1})$ 와는 반대로 점진적으로 noise 를 걷어내는 denoising process $q(x_{t-1}|x_t)$

BUT, 노이즈가 추가된 데이터를 완벽하게 원래 상태로 되돌리는것은 불가능한 일

$$p_\theta(X_{t-1} \mid X_t) \approx q(X_{t-1} \mid X_t)$$
*학습 대상

→ $q(x_{t-1}|x_t)$ 가 아닌 model의 가정을 만족하면서도 $q(x_{t-1}|x_t)$ 와 최대한 유사한 분포 $p_\theta(x_{t-1}|x_t)$ 를 찾는다.

LDMs : noised latent representation z_T 에서 latent representation z로 denoising

Latent Diffusion Models

- **Diffusion Model objective**

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

이 모델은 denoising autoencoder $\epsilon_\theta(x_t,t)$; t=1,…,T의 weighted sequence로 볼 수 있으며, noisy input x_t로 부터 원본 이미지 x를 predict

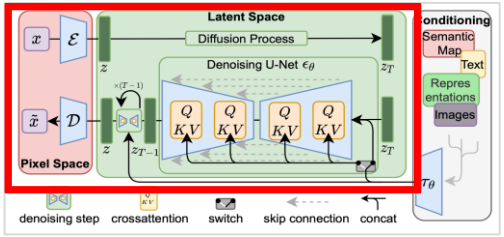
- ϵ : noise, forward diffusion process 를 진행할 때 사용한 실제 noise 값
- ϵ_θ : denoising autoencoder
- x_t : noised sample
- t : noise level, [t-1, t-2,...,0 (less-noisy)]
- $\epsilon_\theta(xt,t)$: denoising process 의 noise

$\boldsymbol{L_{DM}}$: t → t-1, [실제 noise - denoising process 의 noise] 값을 줄여나가는 과정 (loss minimize)

→ 실제 분포와 근사한 분포를 만들어내기 위해 noising process 사용했던 noise ϵ 에 근사시키는 네트워크를 학습하는 것

Latent Diffusion Models

- **Generative Modeling of Latent Representations**



- **Latent Diffusion Model objective**

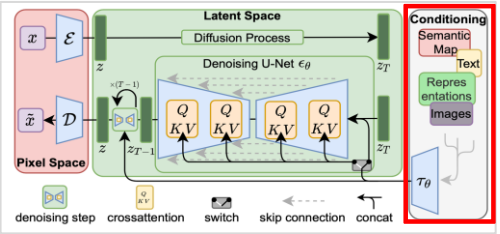
$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right]. \qquad (2)$$

  - denoising model neural backbone $\epsilon_\theta(\circ, t)$: time-conditional UNet
  - x_t : noised sample → z_t : noised latent representation($\varepsilon(x)$)

L_{LDM} **: Diffusion Process($z \rightarrow z_t$)사용했던 noise ϵ 에 근사시키는 네트워크를 학습하는 것**

Conditioning Mechanisms

- **semantic compression**



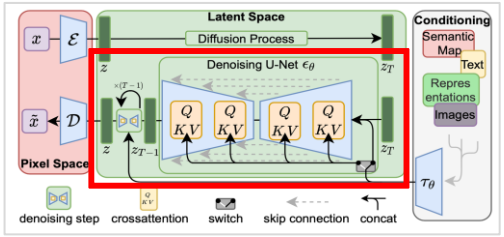
- LDM은 다양한 modalities 적용하기 위해 **conditioning input y** 도입
- diffusion model은 conditional distribution을 p(z|y)로 모델링 가능 → conditional denoising autoencoder $\epsilon_\theta(z_t, t, \mathbf{y})$ 로 구현
  - 이미지 생성을 위한 conditioning input y (text, semantic maps)를 컨트롤하거나, image-to-image translation task를 수행

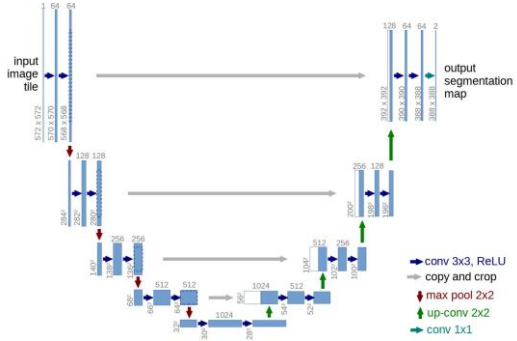
  BUT, conditioning input y를 denoising autoencoder ϵ_θ에 적용하기 위해서는 preprocessing 필요!

- preprocessing[text/image transformer]: **domain specific encoder τ_θ** 도입
  - conditioning input y를 다양한 modalities(language prompts, semantic 맵과 같은)로부터 전처리하기 위해 τ_θ 도입
  - domain specific encoder τ_θ : conditioning input y → intermediate representation $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$ 로 project 한다.
    - **semantic compression:** ϵ_θ의 intermediate layers에 적합하게 encoding

Conditioning Mechanisms

- **conditional denoising autoencoder $\epsilon_\theta(z_t, t, y)$**

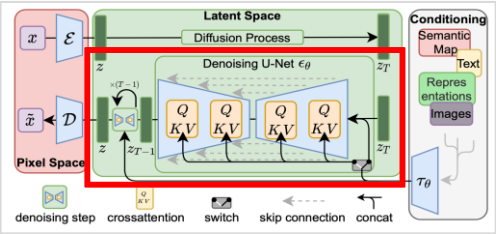


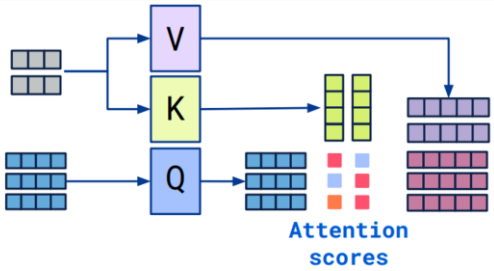


- denoising model neural backbone $\epsilon_\theta(\circ, t)$: **time-conditional UNet structure**
  - skip connection: 인코더 레이어와 디코더 레이어의 직접 연결
  - concatenation: 이미지의 위치와 특징을 추출하기 위해 인코딩 단계의 각 레이어에서 얻은 특징을 디코딩 단계의 각 레이어에 합친다.
- **ϵ_θ intermediate layers**: cross-attention mechanism 적용
  - UNet backbone을 다양한 input modality에 대해 conditioning 할 수 있도록 cross-attention mechanism 으로 구성
    → cross-attention mechanism 을 통해 다양한 input modality 를 model 에 적용

Conditioning Mechanisms

- **conditional denoising autoencoder $\epsilon_\theta(z_t, t, y)$**





- **UNet 의 intermediate layers** : cross-attention layer 로 구성
  - Attention(Q, K, V) = softmax($QK^T/\sqrt{d_k}$)· V가 구현
  - cross attention: 2 개의 embedding sequences 간의 correlation 학습
    - input: Query, Key, Value
    - key, value 의 경우 같은 sequence 에서 얻지만, query는 다른 sequence에서 얻음.(즉, query 출처 ≠ key, value 출처)

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t),\ K = W_K^{(i)} \cdot \tau_\theta(y),\ V = W_V^{(i)} \cdot \tau_\theta(y).$$

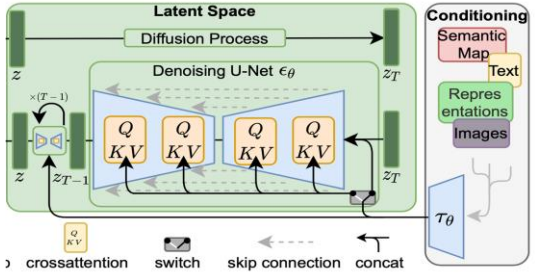
- Q : $\varphi_i(z_t) \in R^{N \times d^i_\epsilon}$: encoder ϵ_θ 을 구현하는 UNet 의 (flattened) intermediate representation
- $W^{(i)}_V \in R^{d \times d^i_\epsilon}$, $W^{(i)}_Q$ & $W^{(i)}_K \in R^{d \times d_\tau}$: learnable projection matrices
- **LDM** cross-attention: Q (intermediate representation) 을 생성할 때, conditioning input y 의 어떤 정보를 Attention 해야할지 파악하는 과정
  → conditioning input y를 참고해서 noised latent representation z_t 를 denoising

Conditioning Mechanisms

- **conditional Latent Diffusion Model objective**

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right], \quad (3)$$

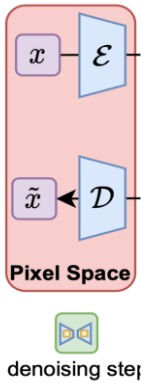
  - τ_θ , ϵ_θ : Eq. 3 을 통해 optimized
  - this conditioning mechanism is flexible



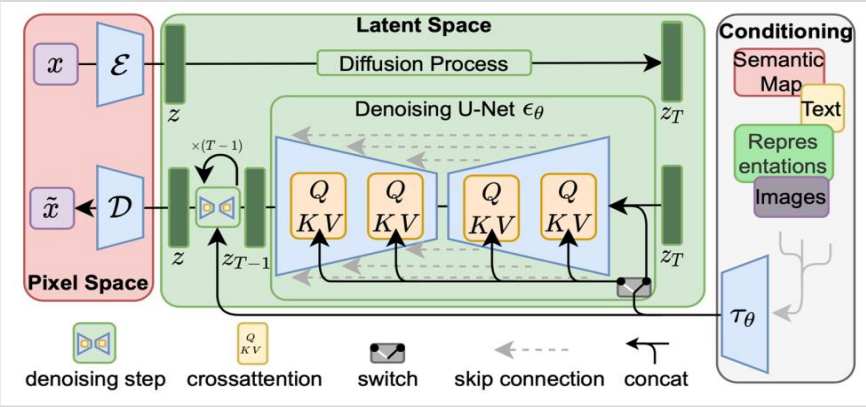
- **autoencoder loss (perceptual loss + a patch-based adversarial objective)**

  - **perceptual loss**: feature map 거리 계산

  - **patch based adversarial objective**: patch 단위로 T/F를 판별하는 방식
    : local realism 실현, (L2 or L1 objectives)처럼 pixel 단위 loss를 사용했을 때 발생하는 blurriness 현상을 완화하여 reconstruction 이 image manifold 에만 국한되도록 보장



LDM process

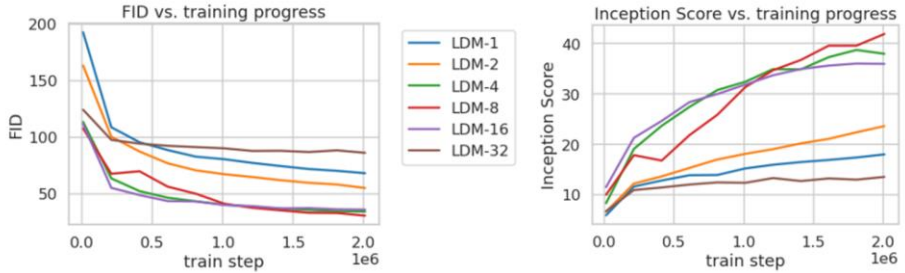


1. **perceptual compression ε**
   : x → latent representation z = ε(x) (latent space에 매핑)
2. **diffusion process**
   : latent representation z → z_T(noised latent representation)
3. **conditioning mechanism & semantic compression τ_θ**
   : conditioning input y → intermediate representation $\tau_\theta(y)$
4. **denoising process**
   : $\tau_\theta(y)$를 참고하여 noised latent representation z_T → latent representation z
5. **perceptual compression D**
   : latent representation z → reconstruction 이미지 $\tilde{x} = D(z)$

Experiments

- **On Perceptual Compression Tradeoffs:** different downsampling factors f 를 가진 LDMs 평가
  - downsampling factors f
    - f ∈ {1, 2, 4, 8, 16, 32} ; LDM-f 로 부름.
    - LDM-1 : pixel-based DMs
  - computational resources
    - a single NVIDIA A100 으로 고정
    - 동일한 steps 과 parameters 개수로 train

❑ **class-conditional LDMs 의 sample quality**



ImageNet dataset 으로 training progress(2M train steps) 를 걸쳐 downsampling factors f 를 적용

- LDM-{4-16} : efficiency and perceptual 간의 좋은 균형을 보임
- small downsampling factors for LDM-{1,2} : slow training progress
- 지나치게 높은 f(LDM-32) : 적은 training steps 후, sample quality 를 제한
  - 강한 perceptual compression 이 information loss 를 일으키고 sample quality 를 제한하기 때문

Experiments

- **Conditional Latent Diffusion - Transformer Encoders for LDMs**
  - text-to-image modeling,
    - LAION-400M 에서 language prompts 에 따라 condition 된 parameter model 을 train
    - BERT-tokenizer 를 사용하고 τ_θ을 transformer 로 구현하여 cross-attention 을 통해 UNet latent code 를 생성한다.
    - [language representation ↔ visual synthesis] 를 학습하기 위한 domain specific encoder τ_θ는 user-defined text prompts 로 일반화

❑ **<Text-To-Image>**

Samples from our text-to-image LDM model for user-defined text prompts, which is trained on LAION-400M.

Experiments

- **Conditional Latent Diffusion - Convolutional Sampling Beyond 256^2**

  - image-to-image translation models
    - spatially 하게 정렬된 conditioning information 을 ϵ_θ input에 concat 함으로써 LDMs 은 image-to-image translation models 구현

  - large resolution
    - input resolution 256^2 (crops from 384^2) 에 대해 train 하지만, 해당 모델은 larger resolutions 으로 일반화되며 convolutional 방식을 사용할 때 megapixel 까지 이미지를 생성할 수 있다.
    - super-resolution models과 inpainting models에 적용하여 512^2 and 1024^2 사이의 이미지 생성

❑ large resolution (256 x 256 → 512×1024)

256^2 resolution 으로 train된 LDM을 풍경 이미지의 conditioned tasks 에 대해 larger resolution (here: 512×1024) 으로 일반화

Experiments

- **Super-Resolution with Latent Diffusion**
  - concatenation을 통해 low-resolution image을 직접 conditioning함으로써 super-resolution을 train
  - 실험 방법
    - SR3 based, image degradation을 4×-downsampling 후, bicubic interpolation 으로 수정하고 SR3's data processing pipeline 을 따라 ImageNet 으로 train
    - f = 4 autoencoding model pretrained on OpenImages 사용하고 low-resolution conditioning y 를 UNet 의 inputs 으로 연결

❑ ImageNet 64→256 super-resolution on ImageNet-Val.

- LDM-SR: realistic textures rendering
- SR3: 일관된 fine structures 를 합성할 수 있다.

Experiments

- **Inpainting with Latent Diffusion**
  - inpainting
    - masked regions of an image 을 새로운 컨텐츠로 채우거나 대체하는 작업

❑ Comparison of inpainting performance

Method	40-50% masked		All samples	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
LDM-4 (ours, big, w/ ft)	**9.39**	0.246± 0.042	**1.50**	0.137± 0.080
LDM-4 (ours, big, w/o ft)	12.89	0.257± 0.047	2.40	0.142± 0.085
LDM-4 (ours, w/ attn)	11.87	0.257± 0.042	2.15	0.144± 0.084
LDM-4 (ours, w/o attn)	12.60	0.259± 0.041	2.37	0.145± 0.084
LaMa [85]†	12.31	**0.243**± 0.038	2.23	**0.134**± 0.080
LaMa [85]	12.0	**0.24**	2.21	0.14
CoModGAN [103]	10.4	0.26	1.82	0.15
RegionWise [51]	21.3	0.27	4.75	0.15
DeepFill v2 [100]	22.1	0.28	5.20	0.16
EdgeConnect [57]	30.5	0.28	8.37	0.16

❑ 사용자 preference 연구

User Study	SR on ImageNet		Inpainting on Places	
	Pixel-DM ($f1$)	*LDM-4*	LAMA [85]	*LDM-4*
Task 1: Preference vs GT ↑	16.0%	**30.4%**	13.6%	**21.0%**
Task 2: Preference Score ↑	29.4%	**70.6%**	31.9%	**68.1%**

- Task 1: ground truth, generated image 중 preference 요청
- Task 2: 2개의 generated images 중 preference 요청

Conclusion

- quality 를 저하시키지 않으면서도 denoising diffusion model 의 training and sampling efficiency 크게 향상시킬 수 있는 latent diffusion model 제시

- cross-attention conditioning mechanism 을 기반으로, Conditional Image Synthesis task 에 다른 SOTA 모델들과 비교해도 손색이 없었다.