Stable Diffusion

Medical Imaging & Intelligent Reality Lab.
Convergence Medicine/Radiology

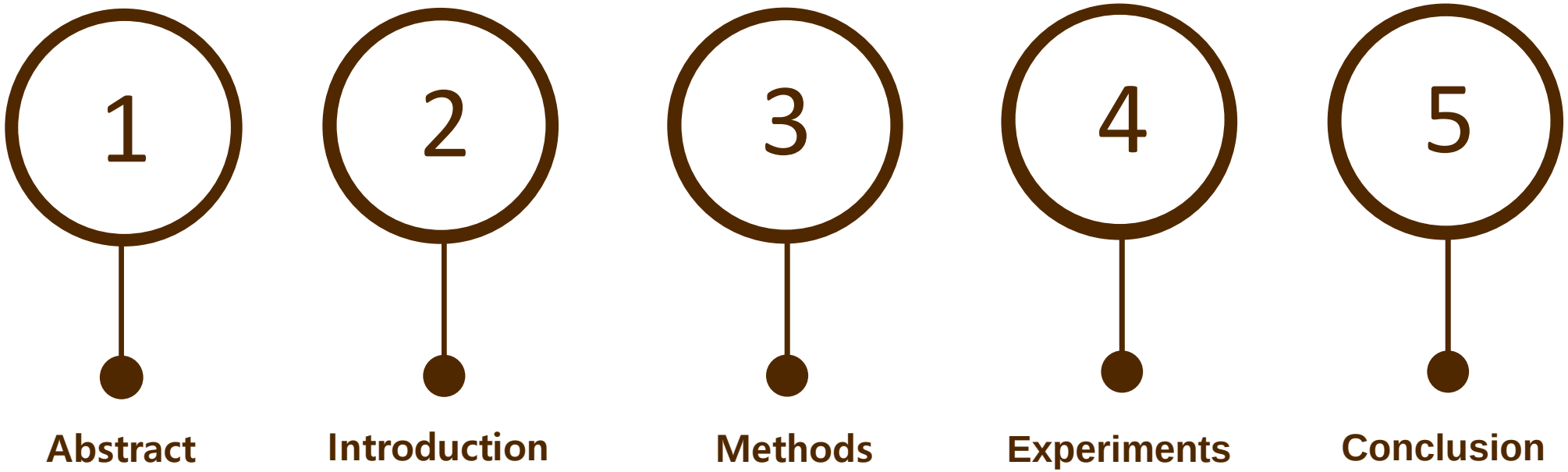
Git: https://githuddb.com/CompVis/latent-diffusion

서울아산병원 의공학연구소

발표자: 경성구

Email: babbu3682@gmail.com

Paper Contents



Background

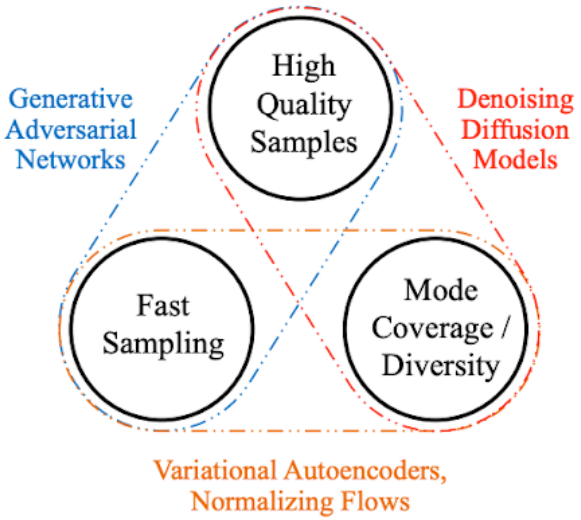
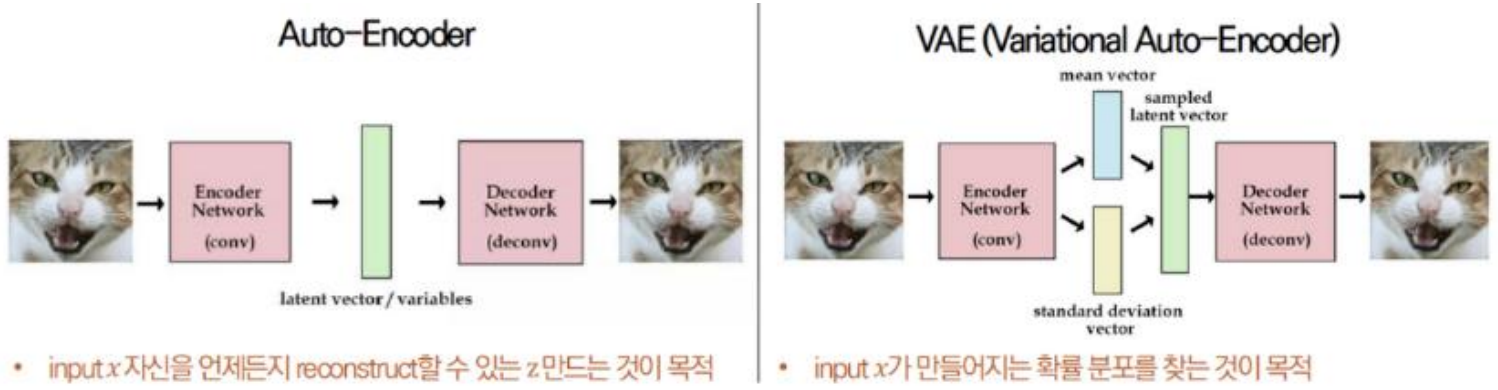


Figure 1. Generative learning trilemma

Background – VAE



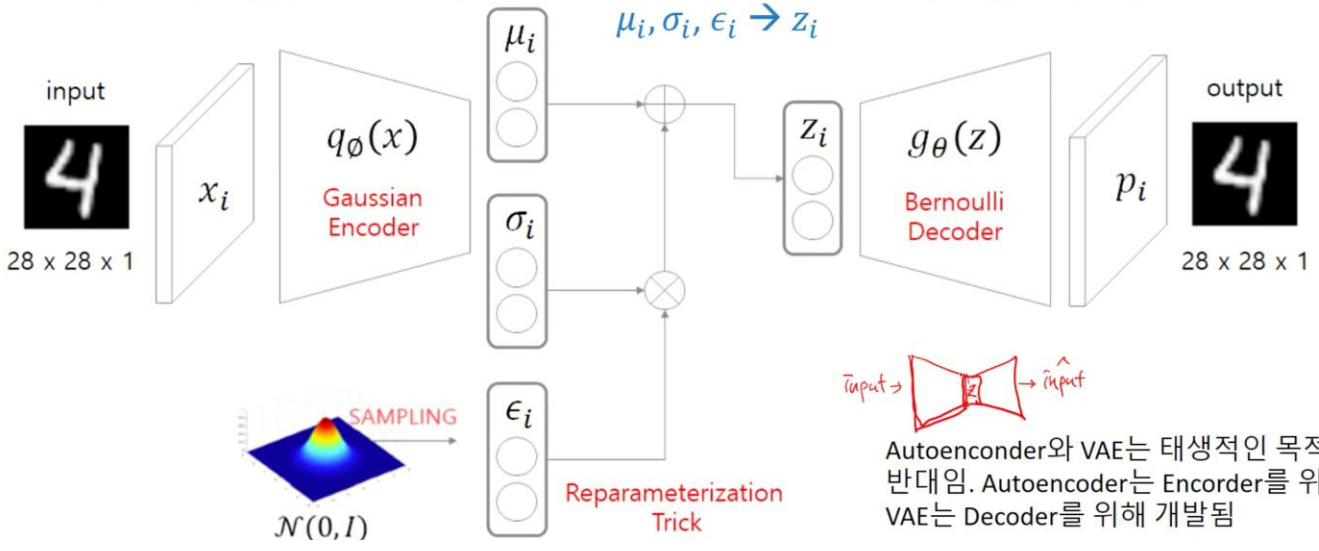
- input x 자신을 언제든지 reconstruct할 수 있는 z 만드는 것이 목적
- input x가 만들어지는 확률 분포를 찾는 것이 목적

Variational Autoencoders (VAE) – How to work

Input: $x_i \rightarrow q_\phi(x) \rightarrow \mu_i, \sigma_i$

$\mu_i, \sigma_i, \epsilon_i \rightarrow z_i$

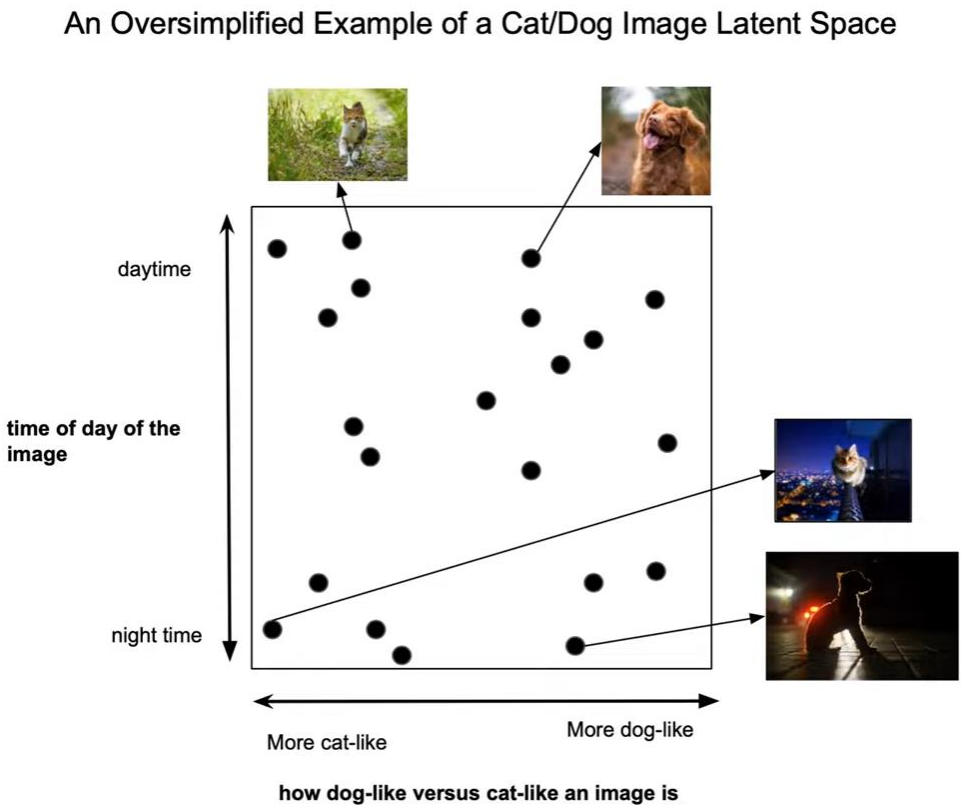
$z_i \rightarrow g_\theta(z) \rightarrow p_i$:output

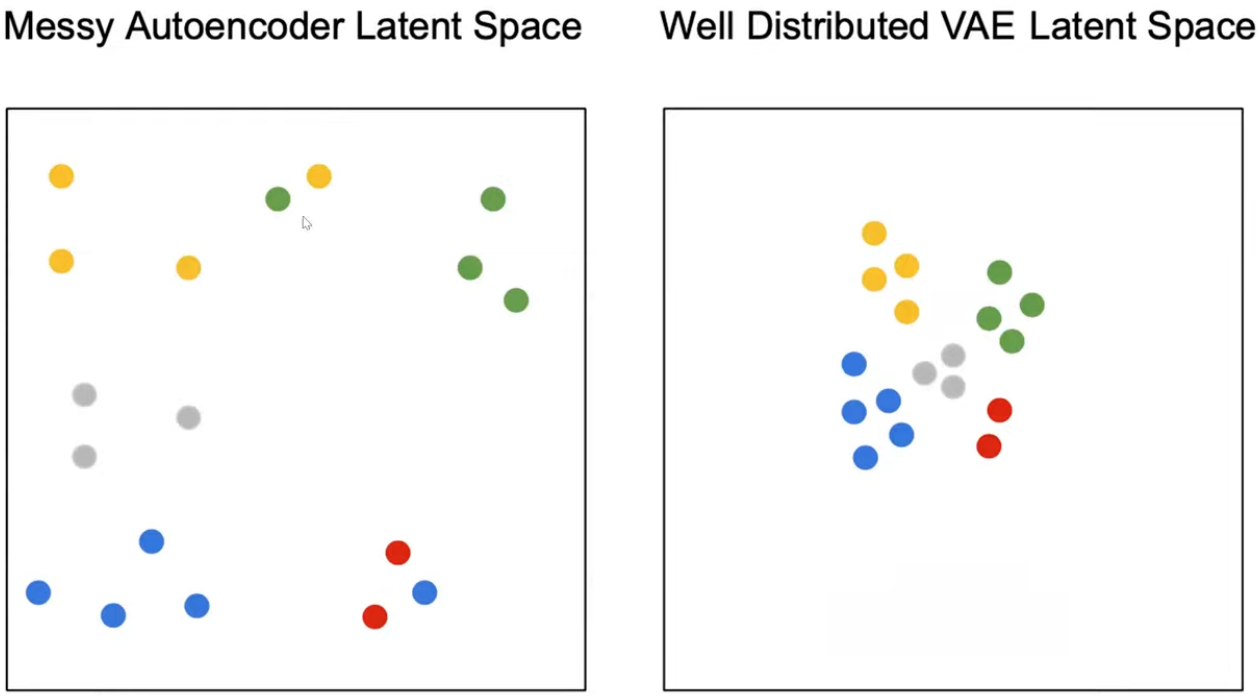


Autoenconder와 VAE는 태생적인 목적이 반대임. Autoencoder는 Encorder를 위해 VAE는 Decoder를 위해 개발됨

$$L = -E_{z\sim q(z|x)}\left[\log p(x|z)\right] + D_{KL}\left(q\left(z|x\right)||p\left(z\right)\right)$$

Background – VAE

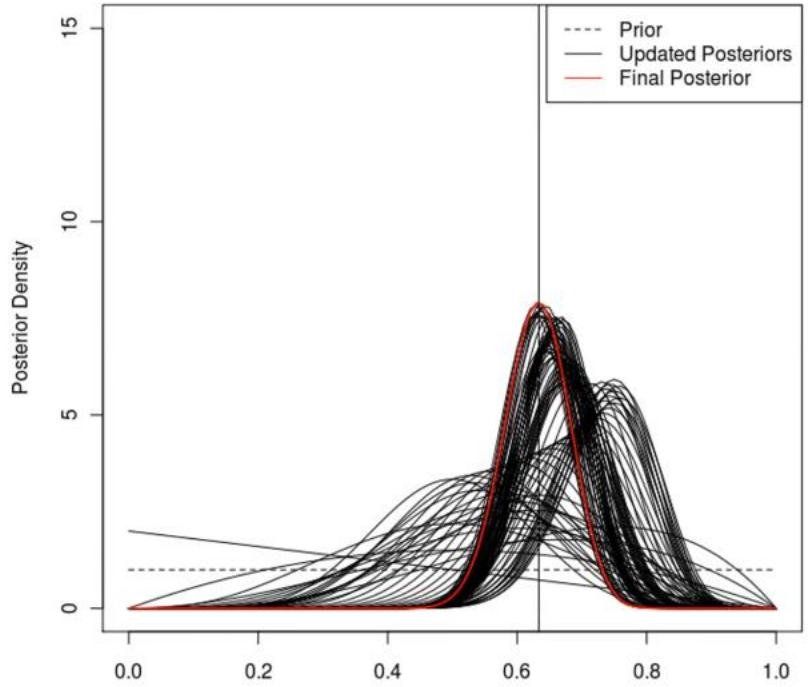




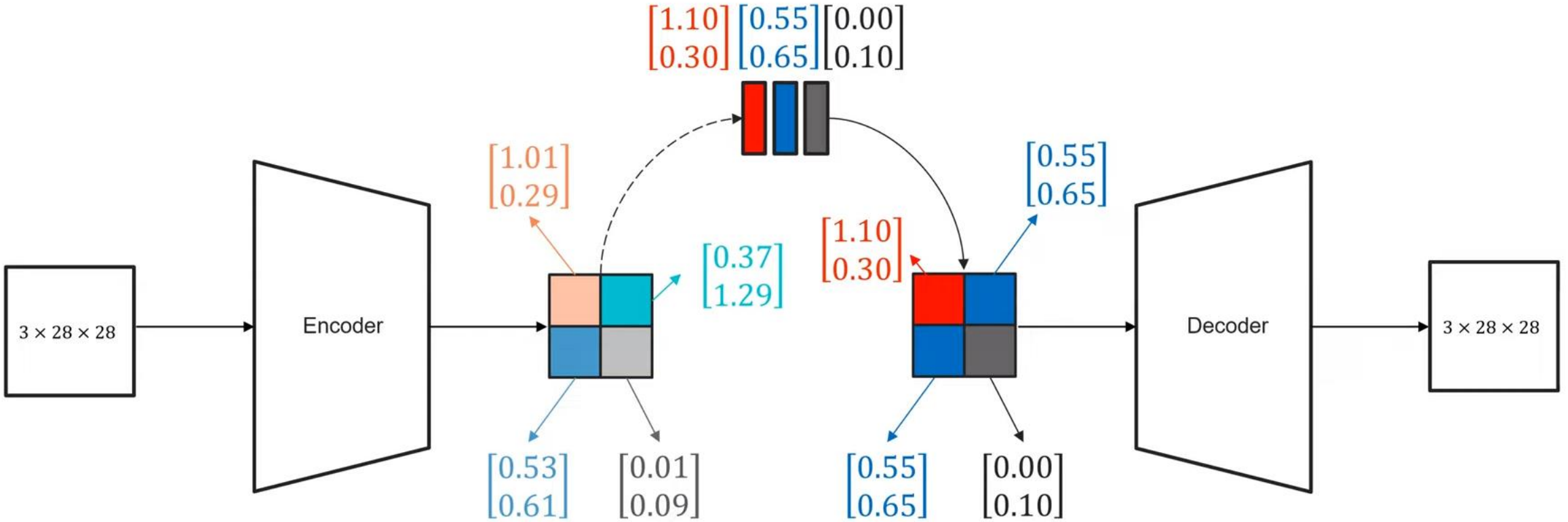
Background – VQ-VAE

Posterior collapse

- 일종의 local minima state
- Approximate posterior가 prior을 그대로 따라함.
- Autoencoder가 latent variable을 무시한 상태에서 학습이 진행됨.
- $q_\phi(z \mid x) = p(z)$



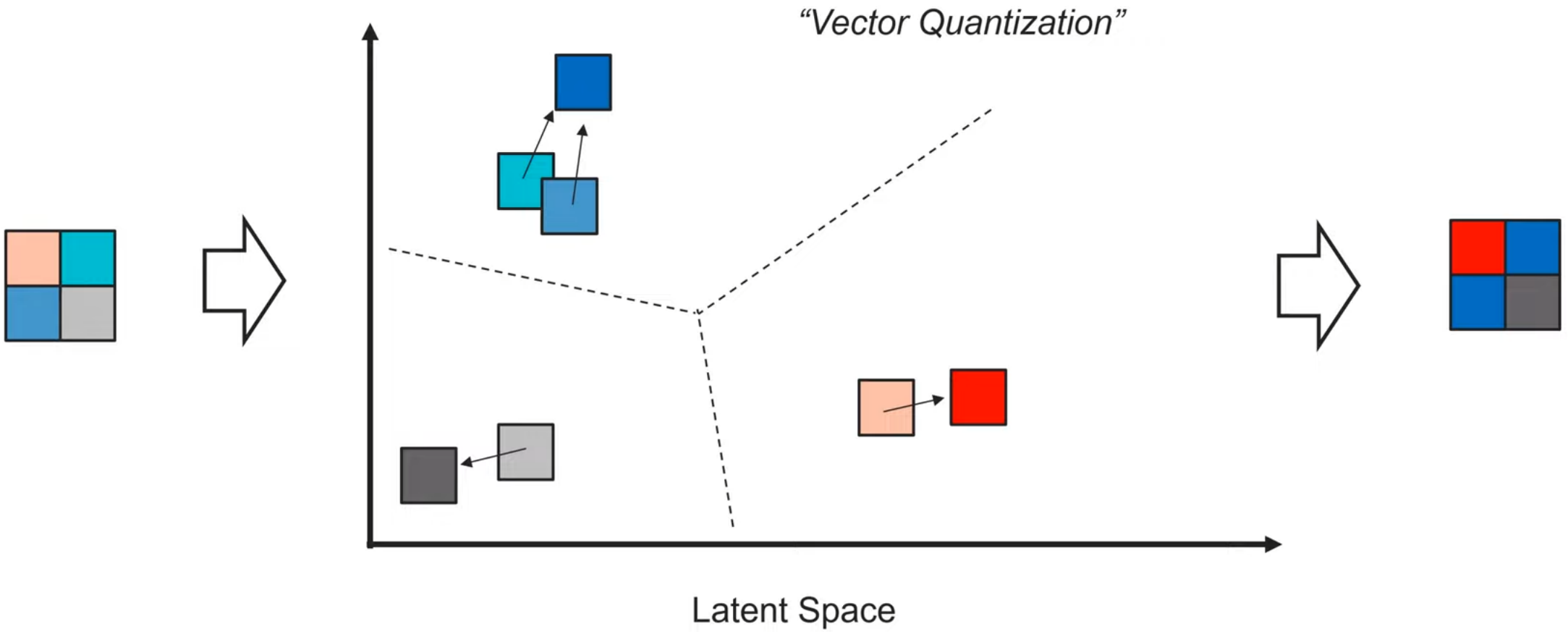
Background – VQ-VAE



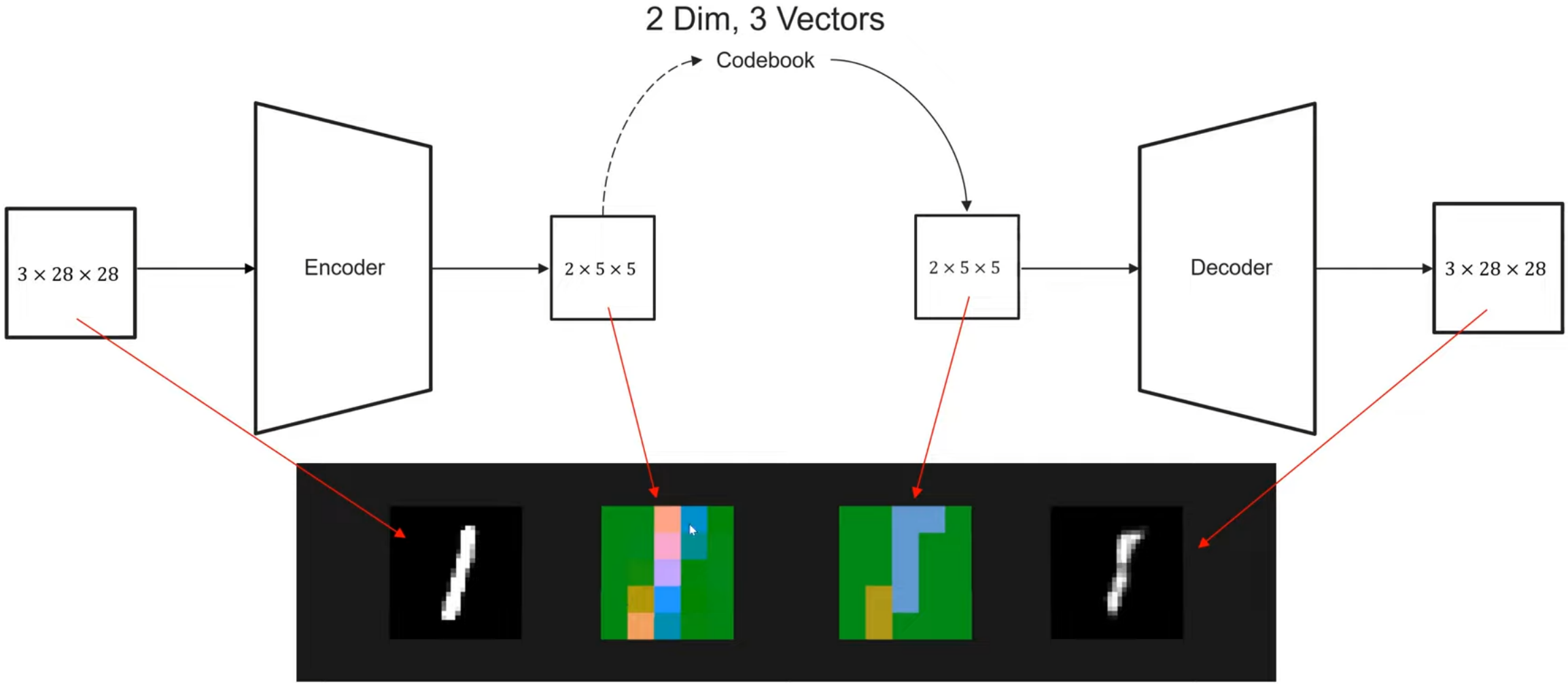
$3^{2\times2}$

Maximum Discrete Code

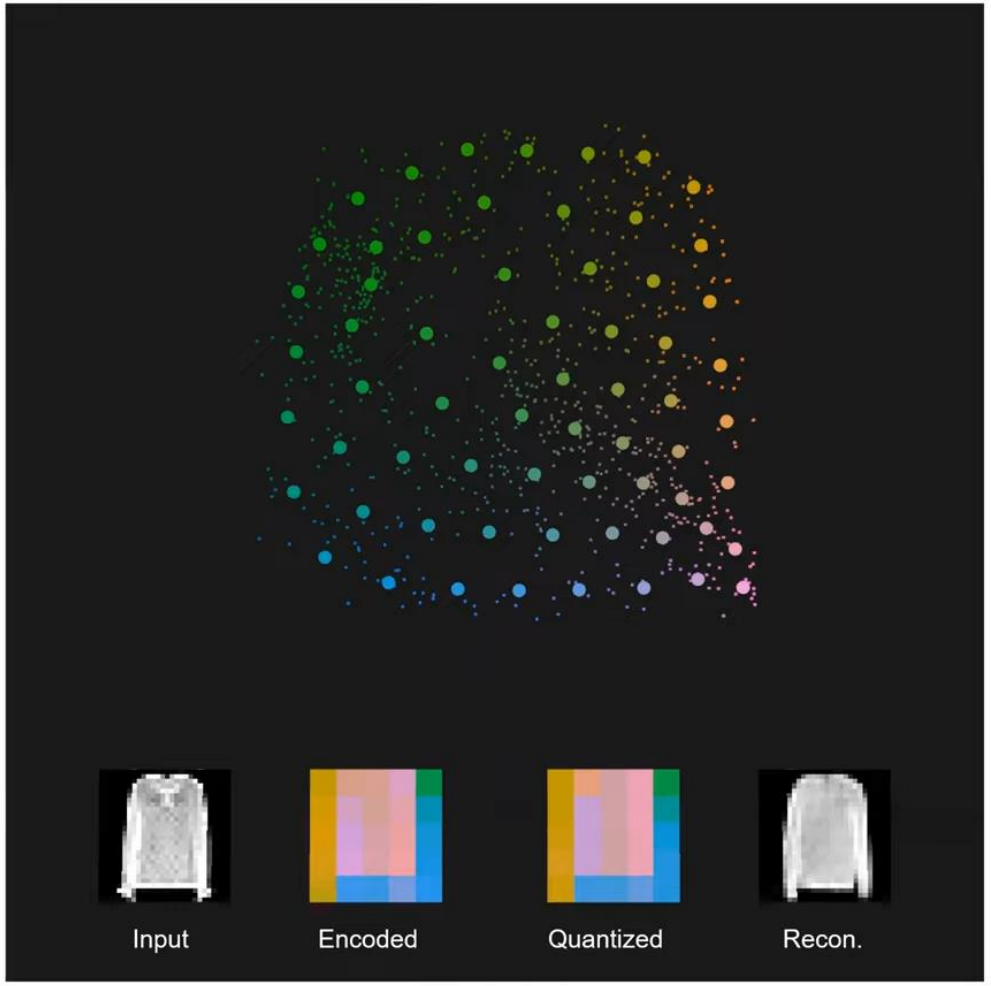
Background – VQ-VAE

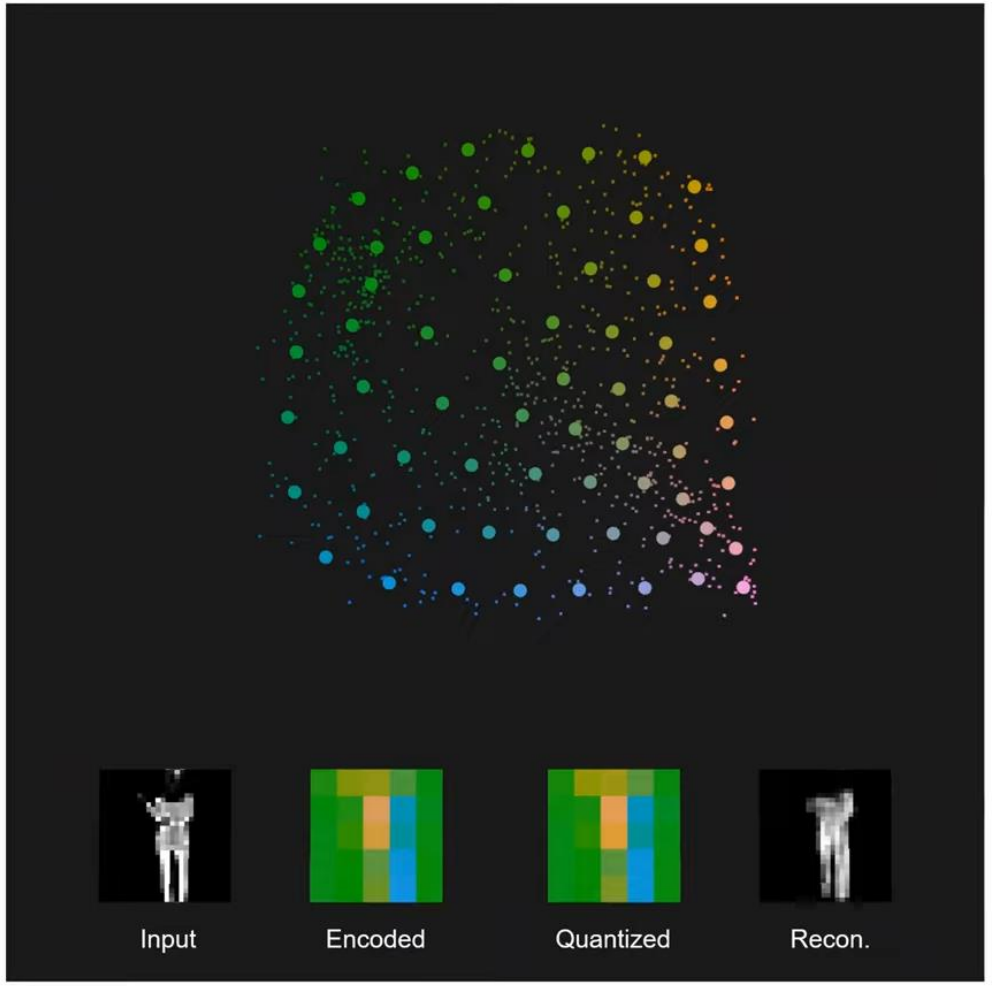


Background – VQ-VAE

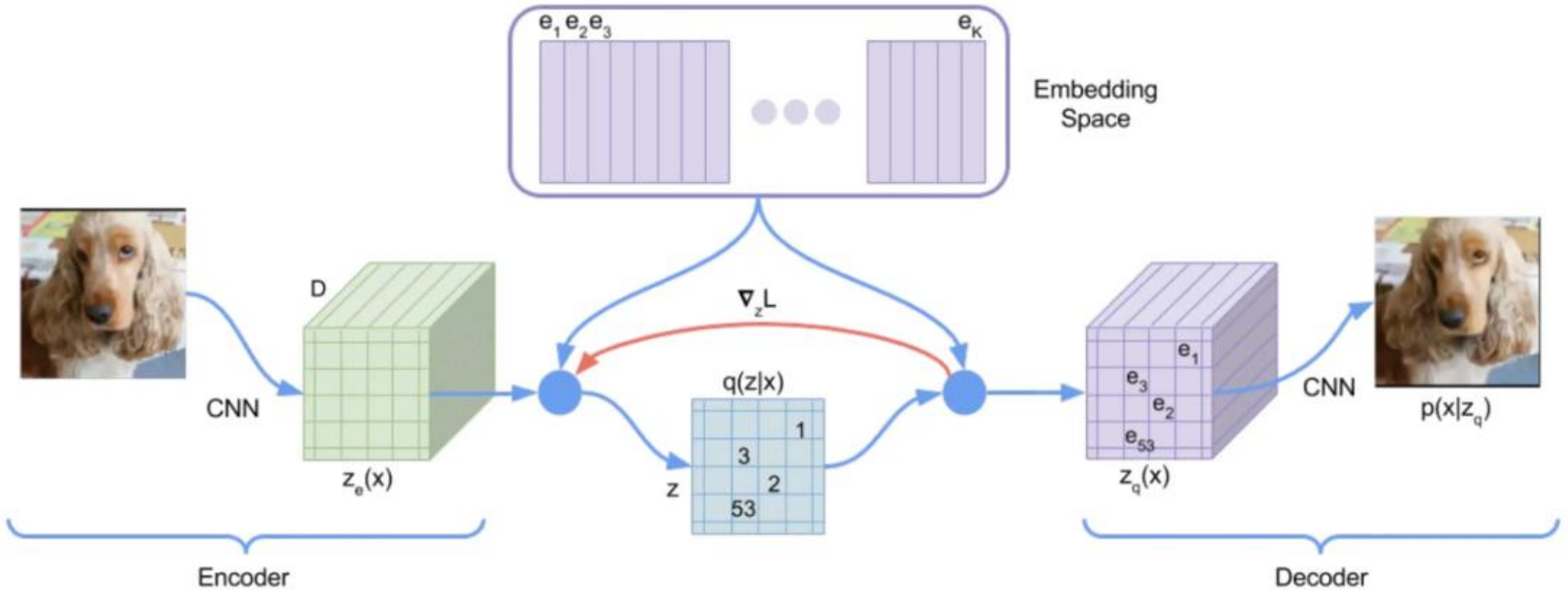


Background – VQ-VAE



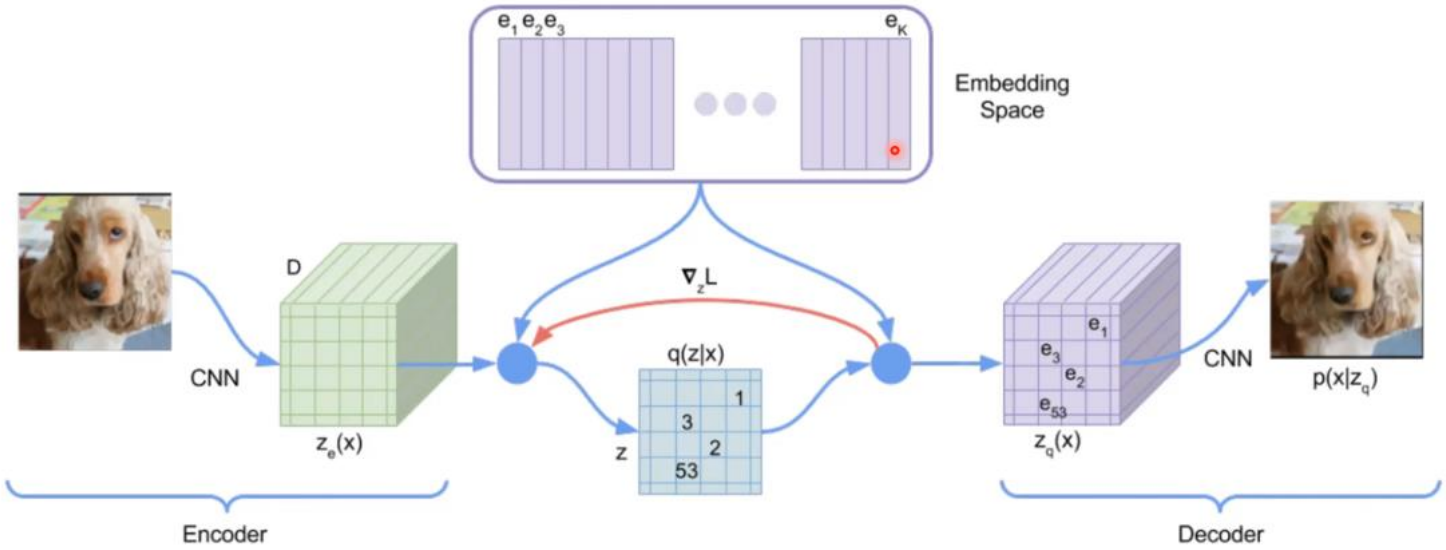


Background – VQ-VAE



- VQ-VAE는 Vector quantization(VQ)을 이용하여 이산 표현을 다룸.
- VQ를 사용할 때, posterior과 prior distribution은 categorical distribution이다.
- 이 분포로부터 생성된 sample은 embedding table을 indexing한다. 이 embedding는 decoder의 입력으로 들어간다.

Background – VQ-VAE



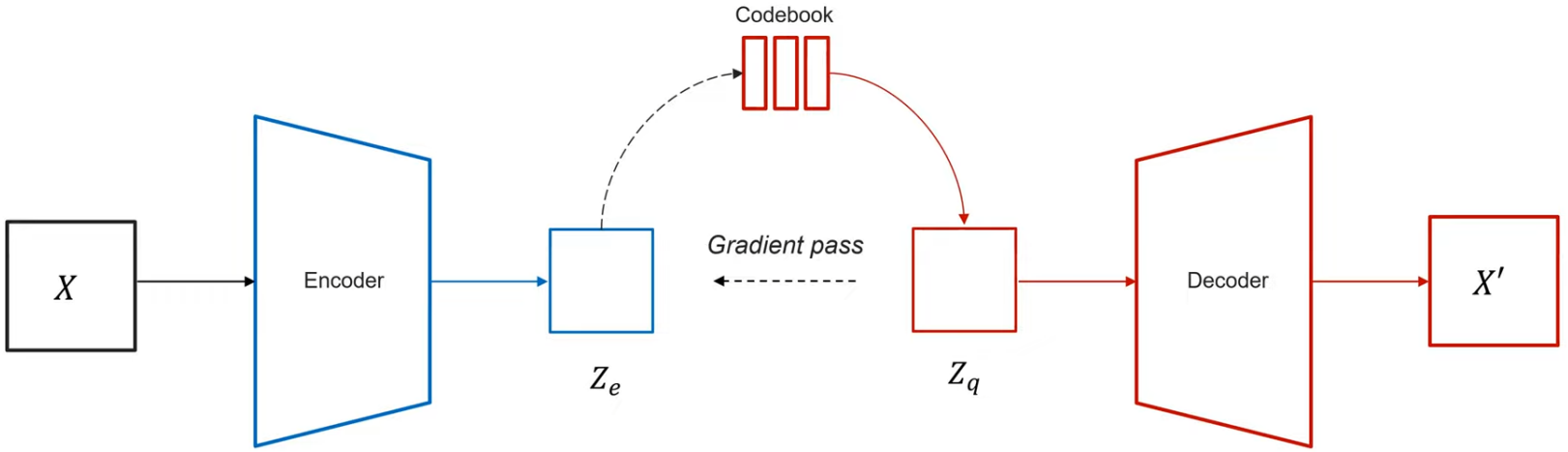
- $e \in R^{R*D}$: latent embedding space(codebook),
  - K: discrete latent space의 크기
  - D: e_i의 차원
- 모델의 encoder는 입력 x 를 받아 discrete latent variable $z_e(x)$ 를 출력
- posterior categorical cistribution q(z|x)의 확률은 크기 K의 one-hot encoding으로 정의됨.

$$q(z=k|x) = \begin{cases} 1 & \text{for } k = \text{argmin}_j \|z_e(x) - e_j\|_2, \\ 0 & \text{otherwise} \end{cases}$$

- decoder의 input $z_q(x)$를 얻기 위해 아래 식과 같이 embedding(codebook) space e에서 $z_e(x)$와 가장 가까운 원소 e_j를 찾는다.

$$z_q(x) = e_k, \quad \text{where} \quad k = \text{argmin}_j \|z_e(x) - e_j\|_2$$

Background – VQ-VAE



$$L = MSE(X, X') + MSE(Z_q, sg[Z_e]) + MSE(sg[Z_q], Z_e)$$
$$= z_e.\text{detach}()$$

- z_q(x)를 계산하는 식(*argmin*)에서는 gradient를 정의할 수 없음.
- z_q로 들어온 gradient를 그대로 z_e에 복사하는 형태로 gradient($\nabla_z L$)를 encoder쪽으로 보내는 방식을 사용했다.

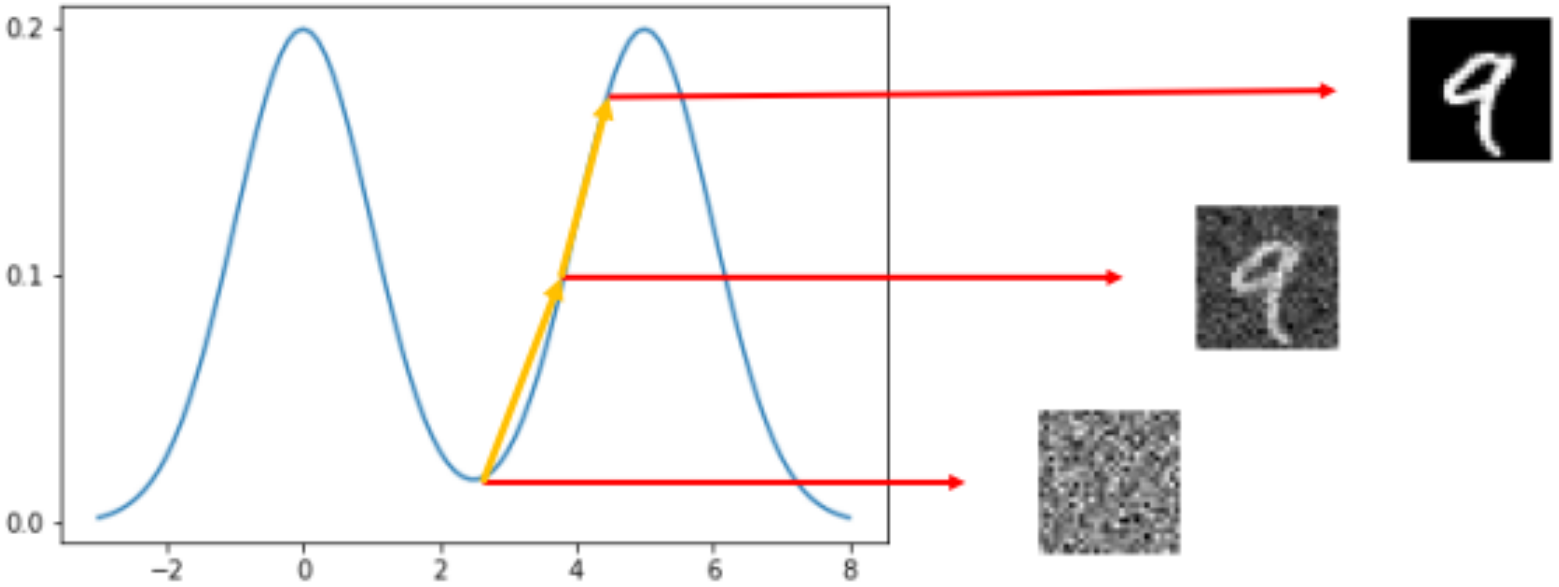
- Loss Function은 아래와 같다.

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2,$$

  - 첫번째 항: reconstruction loss로 encoder, decoder 모두를 최적화
  - 두번째 항: 첫번째 항에서 gradient가 z_q에서 z_e로 바로 넘어가기 때문에 embedding e를 학습시키지 못한다. Encoder로부터 뽑혀 나온 z_e와 비슷해지도록 e를 업데이트
  - 세번째 항: embedding e_i는 encoder parameter만큼 빠르게 학습되지 못한다. 그래서 e와 z_e가 비슷한 속도로 학습될 수 있도록 commitment loss를 넣었다.

Background – DDPM

- 데이터 공간상에서 임의의 데이터 생성 (랜덤 노이즈)
- 확률밀도함수의 기울기를 계산 후 확률 값이 높아지는 방향으로 데이터 업데이트
- 확률 값이 높은 곳에 도달하면 샘플링 된 데이터와 유사한 데이터 생성 가능



Background – DDPM

- **Likelihood(우도)란, 주어진 관측 값이 해당 확률분포에서 나왔을 확률, 주어진 파라미터를 이용한 분포가 모집단의 분포일 확률. 즉, 연속확률밀도함수 *pdf* 의 y axis 값 -> *p(x).***
- **MLE (최대 우도 추정)란, 각 관측 값에 대한 총 가능도가 최대가 되게 하는 분포를 찾는 것.**
- ***pdf* 이므로 모든 sample값을 미분 계산 할 수 없으므로, 평균과 분산의 미분 값이 0이 되는 값을 찾아 MLE가 최대가 되게끔 하는 파라미터 혹은 분포를 찾는다.**

$$Likelihood = \prod p_{\mu,\sigma}(x)$$

$$\frac{\partial Likelihood}{\partial \mu} = 0$$

$$\frac{\partial Likelihood}{\partial \sigma} = 0$$

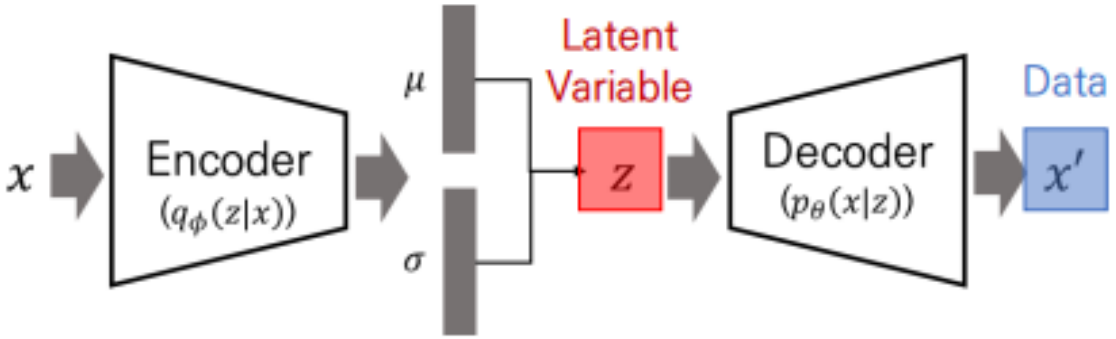
$$\mu_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma_{MLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

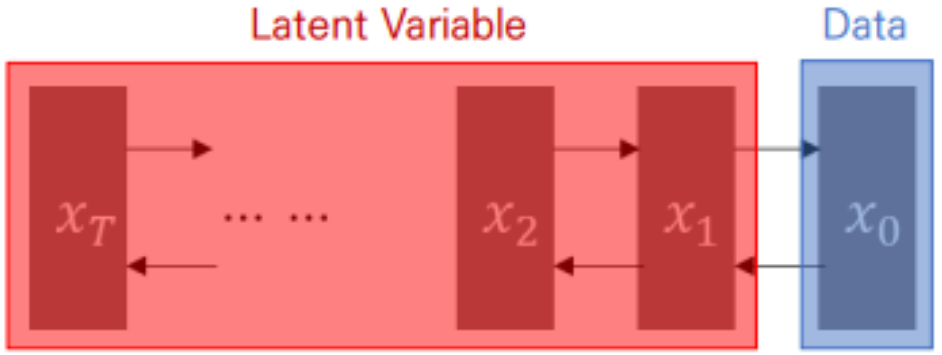
Background – DDPM

Denoising Diffusion Probabilistic Models (DDPM)

- VAE: Latent variable(z)
- Diffusion Models: Latent variable이 Markov chain ($x_1 \sim x_T$)
- Random process: 확률 변수들의 나열
- Markov chain: 이전 시점의 변수에만 영향을 받은 random process ($\mathrm{P}(X_{n+1}|X_n) = \mathrm{P}(X_{n+1}|X_n, X_{n-1}, \dots, X_0)$)

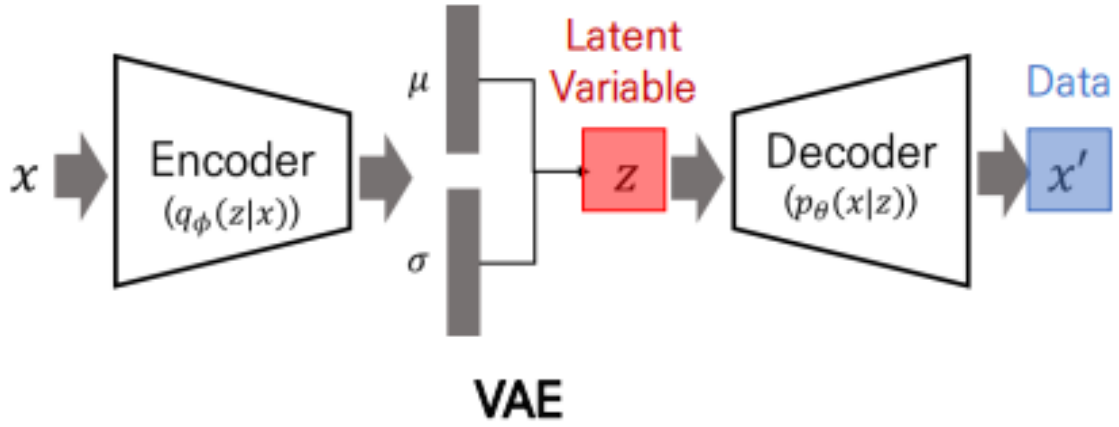


VAE

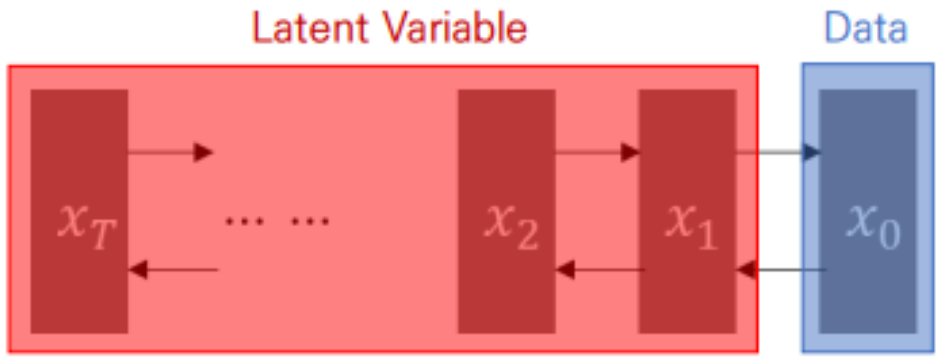


DDPM

Background – DDPM



VAE



DDPM

- Data: $x \rightarrow x_0$
- Latent variable: $\mathbf{z} \rightarrow x_1 \sim x_T (x_{1:T})$

$$Loss_{VAE} = D_{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right) - log p_\theta(x)$$

$$Loss_{DDPM} = D_{KL}\left(q_\phi(x_{1:T}|x_0) \parallel p_\theta(x_{1:T}|x_0)\right) - log p_\theta(x_0)$$

Background – DDPM

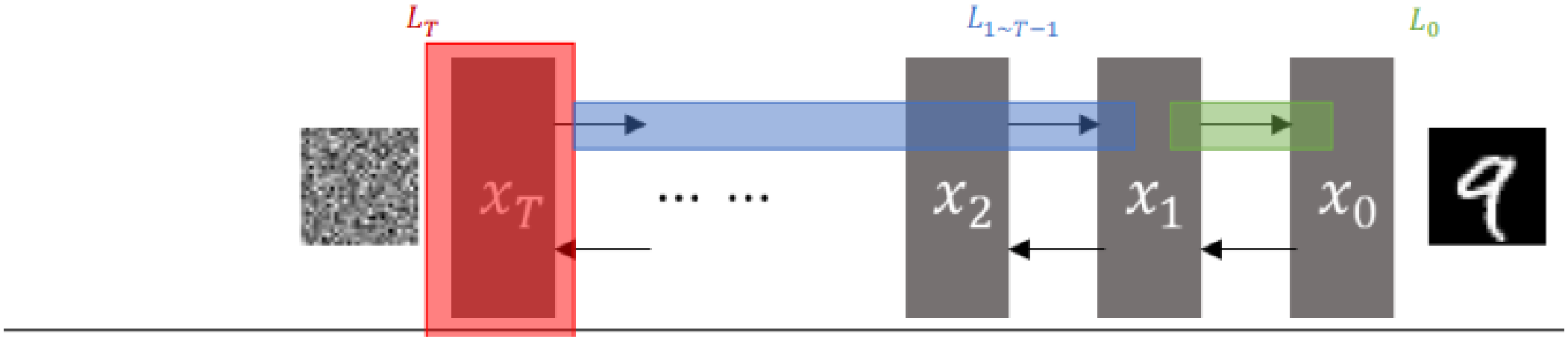
Denoising Diffusion Probabilistic Models (DDPM)

- Forward process: $q(x_t|x_{t-1}) \rightarrow Gaussian$
- Reverse process: $q(x_{t-1}|x_t) \rightarrow Gaussian$

Small noise

- $q(x_{t-1}|x_t, x_0) = \mathrm{N}(x_{t-1};\ \tilde{\mu}(x_t, x_0),\ \tilde{\beta}_t I) \rightarrow True\ Reverse\ Process$
- $p_\theta(x_{t-1}|x_t) \rightarrow Model$

$$Loss_{DDPM} = E_{q_\phi(x_{0:T})}\left[D_{KL}\left(q(x_T|x_0) \parallel p_\theta(x_T)\right) + \sum_{t=2}^{T} D_{KL}\left(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)\right) - log p_\theta(x_0|x_1)\right]$$



High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[R] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [R]Runway ML
https://github.com/CompVis/latent-diffusion

Stable Diffusion

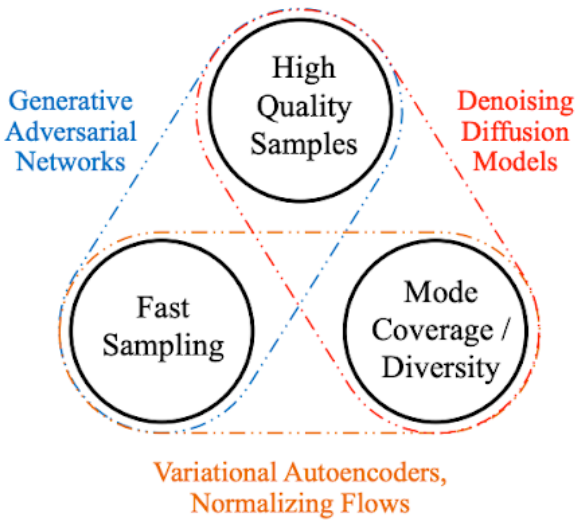


Figure 1. Generative learning trilemma

Abstract

- Since diffusion models typically operate directly in pixel space, optimization of powerful DMs **often consumes hundreds of GPU** days and **inference is expensive** due to sequential evaluations.

- To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in **the latent space of powerful pretrained autoencoders**.

- By introducing **cross-attention layers** into the model architecture, we turn diffusion models into powerful and **flexible generators for general conditioning inputs** (ex. Text, bounding boxes, or high-resolution synthesis becomes possible in a convolutional manner).

- Our latent diffusion models (LDMs) achieve new state-of-the-art scores for **image inpainting** and **class-conditional image synthesis**, including **text-to-image synthesis**, **unconditional image generation** and **super-resolution**, while significantly reducing computational requirements compared to pixel-based DMs

Introduction

- Image synthesis is a field in computer vision that has seen significant development recently but also requires a lot of computational power.
- High-resolution synthesis of complex, natural scenes is typically done using **likelihood-based models with billions of parameters.**
- GANs have shown promising results but **are limited to data with less variability** as their adversarial learning process **does not scale well to complex, multi-modal distributions**.
- Recently, diffusion models, which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive results in this field and are considered state-of-the-art in class-conditional image synthesis and super-resolution.
- Additionally, these models **do not exhibit the same issues** as GANs and **can model complex** natural image distributions **without involving billions of parameters**.

"training such a model requires massive computational resources"

"evaluating an already trained model is also expensive in time and memory"

Introduction

➢ **Democratizing High-Resolution Image Synthesis**

- DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to **spend excessive amounts of capacity** (and thus compute resources) on modeling **imperceptible** details of the data.

- Although the reweighted variational objective aims to address this by under-sampling the initial denoising steps, **DMs are still computationally demanding**, since training and evaluating such **a model requires repeated function evaluations** (and gradient computations) in the high-dimensional space of RGB images.

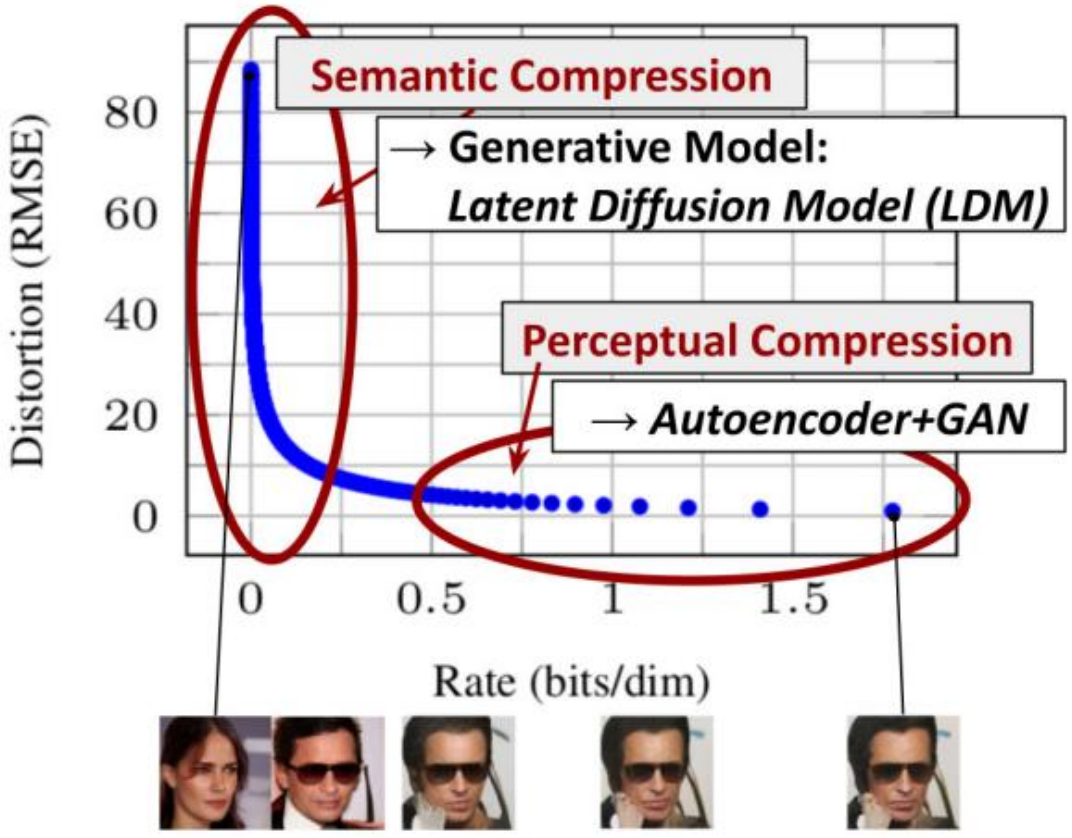
"training such a model requires massive computational resources"

"evaluating an already trained model is also expensive in time and memory"

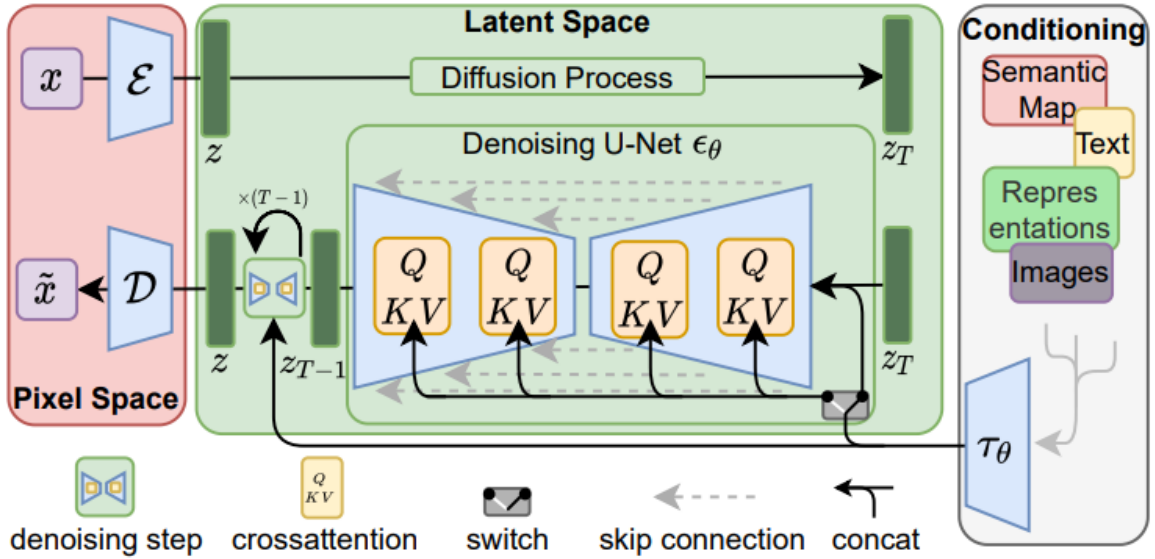
Introduction

➢ **Departure to Latent Space**

- As with any likelihood-based model, learning can be roughly divided into two stages:
- First is a perceptual compression stage which removes high-frequency details(=imperceptible details) but still learns little semantic variation.
- In the second stage, the actual generative model learns the semantic and conceptual composition of the data (semantic compression).
- **We thus aim to first find a perceptually equivalent, but computationally more suitable space**, in which we will train diffusion models for high-resolution image synthesis.



Method



- Utilize an **autoencoding model** which learns a space that is perceptually equivalent to the image space.
- (i) By leaving the high-dimensional image space, we obtain DMs which are computationally much more efficient because sampling is performed on a low-dimensional space.
- (ii) We exploit the inductive bias of DMs inherited from their UNet architecture, which makes them particularly effective for data with spatial structure and therefore alleviates the need for aggressive, quality-reducing compression levels as required by previous approaches.
- (iii) Finally, we obtain general-purpose compression models whose latent space can be used to train multiple generative models and which can also be utilized for other downstream applications

Method

➢ Perceptual Image Compression

- Our perceptual compression model is based on previous work and consists of an autoencoder trained by combination of a perceptual loss and a patch-based adversarial objective.

- This ensures that the reconstructions are confined to the image manifold by enforcing local realism and avoids bluriness introduced by relying solely on pixel-space losses

- the encoder E encodes x into a latent representation $z = E(x)$, and the decoder D reconstructs the image from the latent, giving $\tilde{x} = D(z) = D(E(x))$, where $z \in R^{h \times w \times c}$

- Importantly, the encoder downsamples the image by a factor $f = H/h = W/w$, and we investigate different downsampling factors $f = 2^m$, with $m \in N$.

- KL-reg, VQ-reg

Method

➢ Generative Modeling of Latent Representations

- We now have access to an efficient, low-dimensional latent space in which high-frequency, imperceptible details are abstracted away.

- Compared to the high-dimensional pixel space, this space is more suitable for likelihood-based generative models, as they can now (i) focus on the important, semantic bits of the data and (ii) train in a lower dimensional, computationally much more efficient space

- the ability to build the underlying UNet primarily from 2D convolutional layers, and further focusing the objective on the perceptually most relevant bits using the reweighted bound, which now reads the neural backbone $E_\theta(\circ, t)$ of our model is realized as a time-conditional UNet

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

Method

➢ Conditioning Mechanisms

- This can be implemented with a conditional denoising autoencoder θ(zt, t, y) and paves the way to controlling the synthesis process through inputs y such as text [68], semantic maps [33, 61] or other image-to-image translation tasks [34].

- We turn DMs into more flexible conditional image generators by augmenting their underlying UNet backbone with the cross-attention mechanism

- we introduce a domain specific encoder τθ that projects y to an intermediate representation $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$, which is then mapped to the intermediate layers of the UNet via a cross-attention layer

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \text{ with}$$

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

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

where both τ_θ and ϵ_θ are jointly optimized via Eq. 3.

Experiments

- **Quantitative Results in Unconditional and Class-Conditional Image Synthesis**
  - the FID-, Precision- and Recall-scores (torch-fidelity package)
  - 50k samples from our models and the entire training set of each of the shown datasets
- **Text-to-Image Synthesis**
  - FID and Inception Score (torch-fidelity package)
  - 30000 samples from the validation set of the MS-COCO dataset
- **Layout-to-Image Synthesis**
  - FID score (torch-fidelity package)
  - we similarly follow their protocol and use 2048 center-cropped test images from the validation set.
- **Super Resolution**
  - FID score (torch-fidelity package)
  - images with a shorter size less than 256 px are removed, the low-resolution images are produced using bicubic interpolation with anti-aliasing.
- **User Study**
  - 'Which of the two images is a better high quality version of the low resolution image in the middle?'
  - 'Which of the two images contains more realistic inpainted regions of the image in the middle?'

Experiments

- **Hyperparameters**

We provide an overview of the hyperparameters of all trained *LDM* models in Tab. 12, Tab. 13, Tab. 14 and Tab. 15.

	CelebA-HQ 256×256	FFHQ 256×256	LSUN-Churches 256×256	LSUN-Bedrooms 256×256
f	4	4	8	4
z-shape	$64 \times 64 \times 3$	$64 \times 64 \times 3$	-	$64 \times 64 \times 3$
$\lvert\mathcal{Z}\rvert$	8192	8192	-	8192
Diffusion steps	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear
Nparams	274M	274M	294M	274M
Channels	224	224	192	224
Depth	2	2	2	2
Channel Multiplier	1,2,3,4	1,2,3,4	1,2,2,4,4	1,2,3,4
Attention resolutions	32, 16, 8	32, 16, 8	32, 16, 8, 4	32, 16, 8
Head Channels	32	32	24	32
Batch Size	48	42	96	48
Iterations*	410k	635k	500k	1.9M
Learning Rate	9.6e-5	8.4e-5	5.e-5	9.6e-5

Table 12. Hyperparameters for the unconditional *LDMs* producing the numbers shown in Tab. 1. All models trained on a single NVIDIA A100.

Experiments

- **Hyperparameters**

	LDM-1	*LDM-2*	*LDM-4*	*LDM-8*	*LDM-16*	*LDM-32*
z-shape	$256 \times 256 \times 3$	$128 \times 128 \times 2$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$16 \times 16 \times 8$	$88 \times 8 \times 32$
$\lvert\mathcal{Z}\rvert$	-	2048	8192	16384	16384	16384
Diffusion steps	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear
Model Size	396M	391M	391M	395M	395M	395M
Channels	192	192	192	256	256	256
Depth	2	2	2	2	2	2
Channel Multiplier	1,1,2,2,4,4	1,2,2,4,4	1,2,3,5	1,2,4	1,2,4	1,2,4
Number of Heads	1	1	1	1	1	1
Batch Size	7	9	40	64	112	112
Iterations	2M	2M	2M	2M	2M	2M
Learning Rate	4.9e-5	6.3e-5	8e-5	6.4e-5	4.5e-5	4.5e-5
Conditioning	CA	CA	CA	CA	CA	CA
CA-resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	16, 8, 4	8, 4, 2
Embedding Dimension	512	512	512	512	512	512
Transformers Depth	1	1	1	1	1	1

Table 13. Hyperparameters for the conditional *LDMs* trained on the ImageNet dataset for the analysis in Sec. 4.1. All models trained on a single NVIDIA A100.

Experiments

- **Hyperparameters**

Task	Text-to-Image	Layout-to-Image		Class-Label-to-Image	Super Resolution	Inpainting	Semantic-Map-to-Image
Dataset	LAION	OpenImages	COCO	ImageNet	ImageNet	Places	Landscapes
f	8	4	8	4	4	4	8
z-shape	$32 \times 32 \times 4$	$64 \times 64 \times 3$	$32 \times 32 \times 4$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$64 \times 64 \times 3$	$32 \times 32 \times 4$
$\|\mathcal{Z}\|$	-	8192	16384	8192	8192	8192	16384
Diffusion steps	1000	1000	1000	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear	linear	linear	linear
Model Size	1.45B	306M	345M	395M	169M	215M	215M
Channels	320	128	192	192	160	128	128
Depth	2	2	2	2	2	2	2
Channel Multiplier	1,2,4,4	1,2,3,4	1,2,4	1,2,3,5	1,2,2,4	1,4,8	1,4,8
Number of Heads	8	1	1	1	1	1	1
Dropout	-	-	0.1	-	-	-	-
Batch Size	680	24	48	1200	64	128	48
Iterations	390K	4.4M	170K	178K	860K	360K	360K
Learning Rate	1.0e-4	4.8e-5	4.8e-5	1.0e-4	6.4e-5	1.0e-6	4.8e-5
Conditioning	CA	CA	CA	CA	concat	concat	concat
(C)A-resolutions	32, 16, 8	32, 16, 8	32, 16, 8	32, 16, 8	-	-	-
Embedding Dimension	1280	512	512	512	-	-	-
Transformer Depth	1	3	2	1	-	-	-

Table 15. Hyperparameters for the conditional *LDMs* from Sec. 4. All models trained on a single NVIDIA A100 except for the inpainting model which was trained on eight V100.

Experiments

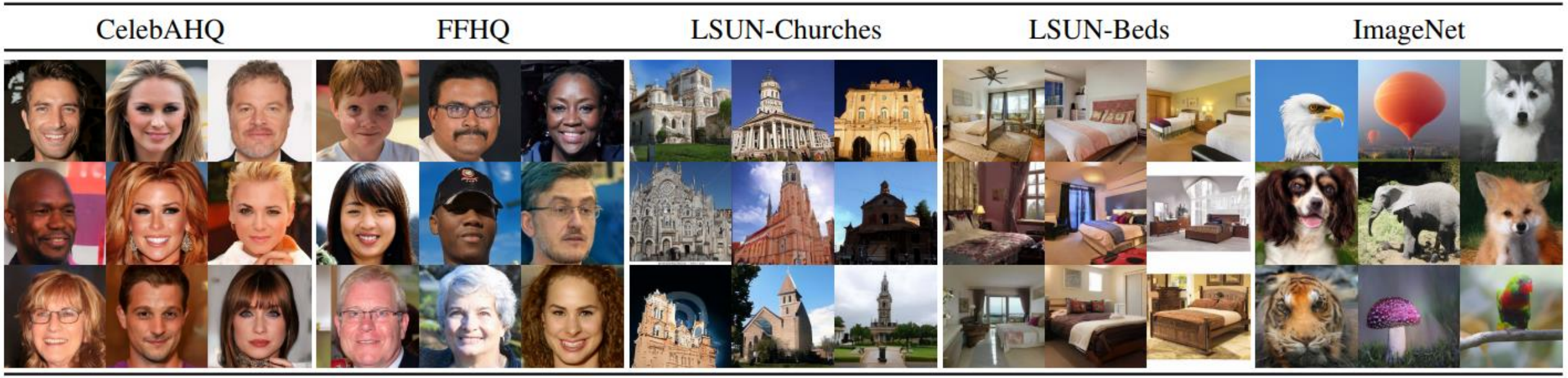


Figure 4. Samples from *LDMs* trained on CelebAHQ [39], FFHQ [41], LSUN-Churches [102], LSUN-Bedrooms [102] and class-conditional ImageNet [12], each with a resolution of 256 × 256. Best viewed when zoomed in. For more samples *cf*. the supplement.

Experiments

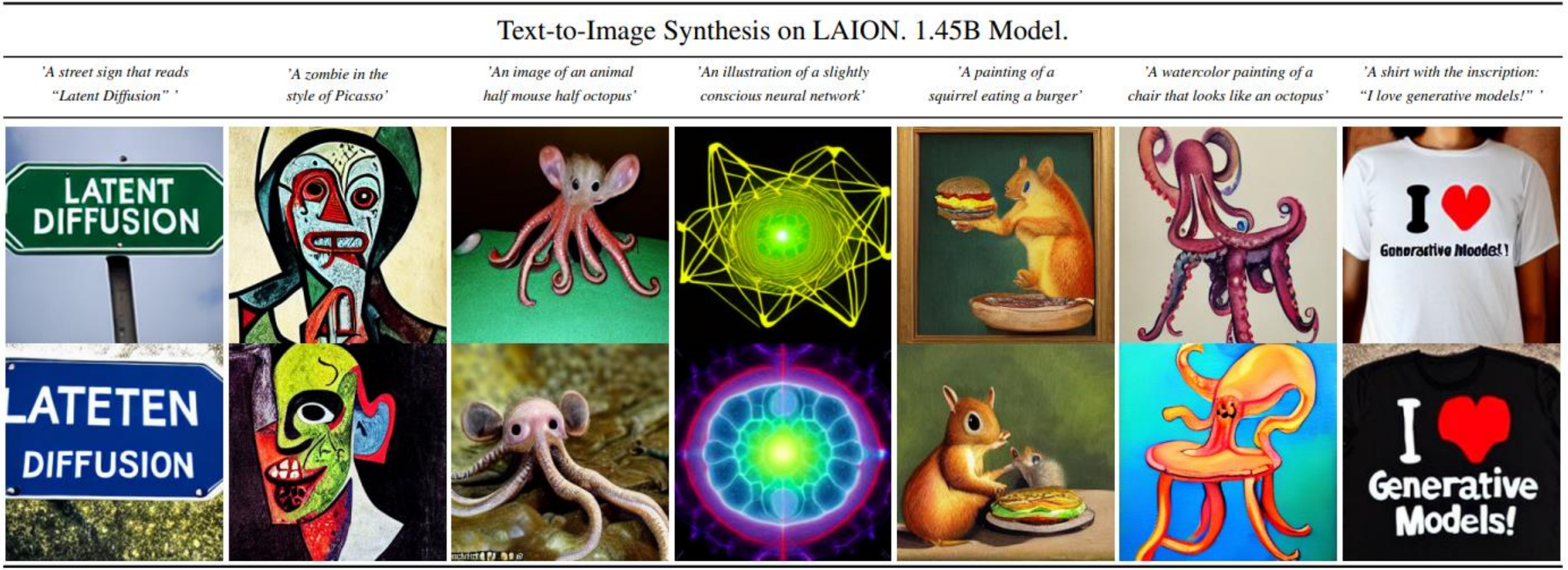


Figure 5. Samples for user-defined text prompts from our model for text-to-image synthesis, *LDM-8 (KL)*, which was trained on the LAION [78] database. Samples generated with 200 DDIM steps and $\eta = 1.0$. We use unconditional guidance [32] with $s = 10.0$.

Experiments

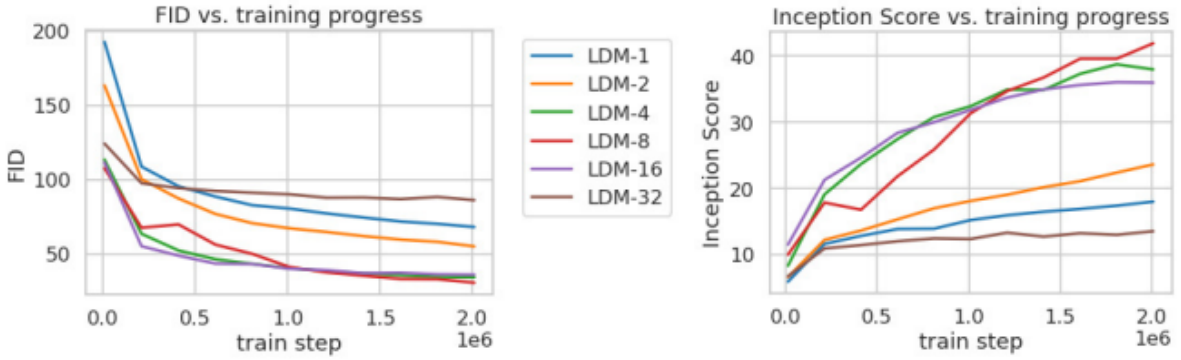


Figure 6. Analyzing the training of class-conditional *LDMs* with different downsampling factors f over 2M train steps on the ImageNet dataset. Pixel-based *LDM-1* requires substantially larger train times compared to models with larger downsampling factors (*LDM-*$\{$*4-16*$\}$). Too much perceptual compression as in *LDM-32* limits the overall sample quality. All models are trained on a single NVIDIA A100 with the same computational budget. Results obtained with 100 DDIM steps [84] and $\kappa = 0$.

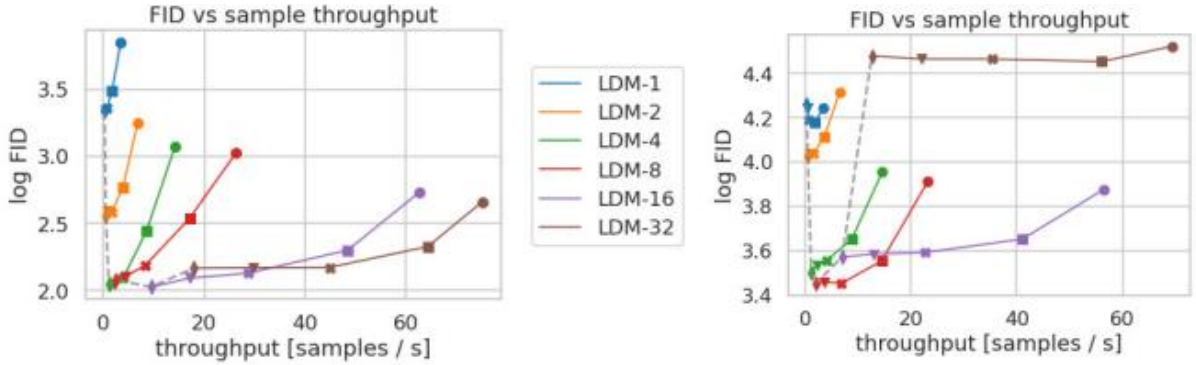


Figure 7. Comparing *LDMs* with varying compression on the CelebA-HQ (left) and ImageNet (right) datasets. Different markers indicate $\{10, 20, 50, 100, 200\}$ sampling steps using DDIM, from right to left along each line. The dashed line shows the FID scores for 200 steps, indicating the strong performance of *LDM-*$\{$*4-8*$\}$. FID scores assessed on 5000 samples. All models were trained for 500k (CelebA) / 2M (ImageNet) steps on an A100.

Experiments

CelebA-HQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
DC-VAE [63]	15.8	-	-
VQGAN+T. [23] (k=400)	10.2	-	-
PGGAN [39]	8.0	-	-
LSGM [93]	7.22	-	-
UDM [43]	7.16	-	-
LDM-4 (ours, 500-s†)	**5.11**	0.72	0.49

FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
ImageBART [21]	9.57	-	-
U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
UDM [43]	5.54	-	-
StyleGAN [41]	4.16	0.71	0.46
ProjectedGAN [76]	**3.08**	0.65	0.46
LDM-4 (ours, 200-s)	4.98	**0.73**	**0.50**

LSUN-Churches 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
DDPM [30]	7.89	-	-
ImageBART [21]	7.32	-	-
PGGAN [39]	6.42	-	-
StyleGAN [41]	4.21	-	-
StyleGAN2 [42]	3.86	-	-
ProjectedGAN [76]	**1.59**	0.61	0.44
*LDM-8** (ours, 200-s)	4.02	**0.64**	**0.52**

LSUN-Bedrooms 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
ImageBART [21]	5.51	-	-
DDPM [30]	4.9	-	-
UDM [43]	4.57	-	-
StyleGAN [41]	2.35	0.59	0.48
ADM [15]	1.90	**0.66**	**0.51**
ProjectedGAN [76]	**1.52**	0.61	0.34
LDM-4 (ours, 200-s)	2.95	**0.66**	0.48

Table 1. Evaluation metrics for unconditional image synthesis. CelebA-HQ results reproduced from [43, 63, 100], FFHQ from [42, 43]. †: N-s refers to N sampling steps with the DDIM [84] sampler. *: trained in KL-regularized latent space. Additional results can be found in the supplementary.

Text-Conditional Image Synthesis				
Method	FID ↓	IS↑	Nparams	
CogView† [17]	27.10	18.20	4B	self-ranking, rejection rate 0.017
LAFITE† [109]	26.94	26.02	75M	
GLIDE* [59]	12.24	-	6B	277 DDIM steps, c.f.g. [32] $s = 3$
Make-A-Scene* [26]	**11.84**	-	4B	c.f.g for AR models [98] $s = 5$
LDM-KL-8	23.31	$20.03_{\pm 0.33}$	1.45B	250 DDIM steps
*LDM-KL-8-G**	12.63	$\mathbf{30.29}_{\pm \mathbf{0.42}}$	1.45B	250 DDIM steps, c.f.g. [32] $s = 1.5$

Table 2. Evaluation of text-conditional image synthesis on the 256 × 256-sized MS-COCO [51] dataset: with 250 DDIM [84] steps our model is on par with the most recent diffusion [59] and autoregressive [26] methods despite using significantly less parameters. †/*:Numbers from [109]/ [26]

Experiments

Figure 8. Layout-to-image synthesis with an *LDM* on COCO [4], see Sec. 4.3.1. Quantitative evaluation in the supplement D.3.

Experiments

Method	FID↓	IS↑	Precision↑	Recall↑	N_{params}	
BigGan-deep [3]	6.95	$203.6_{\pm 2.6}$	**0.87**	0.28	340M	-
ADM [15]	10.94	100.98	0.69	**0.63**	554M	250 DDIM steps
ADM-G [15]	4.59	186.7	0.82	0.52	608M	250 DDIM steps
LDM-4 (ours)	10.56	$103.49_{\pm 1.24}$	0.71	0.62	400M	250 DDIM steps
LDM-4-G (ours)	**3.60**	$\mathbf{247.67}_{\pm 5.59}$	**0.87**	0.48	400M	250 steps, c.f.g [32], $s = 1.5$

Table 3. Comparison of a class-conditional ImageNet *LDM* with recent state-of-the-art methods for class-conditional image generation on ImageNet [12]. A more detailed comparison with additional baselines can be found in D.4, Tab. 10 and F. *c.f.g.* denotes classifier-free guidance with a scale s as proposed in [32].

Experiments

Figure 9. A *LDM* trained on 256^2 resolution can generalize to larger resolution (here: 512×1024) for spatially conditioned tasks such as semantic synthesis of landscape images. See Sec. 4.3.2.

Experiments

Figure 10. ImageNet 64→256 super-resolution on ImageNet-Val. *LDM-SR* has advantages at rendering realistic textures but SR3 can synthesize more coherent fine structures. See appendix for additional samples and cropouts. SR3 results from [72].

Experiments

	SR on ImageNet		Inpainting on Places	
User Study	Pixel-DM ($f1$)	*LDM-4*	LAMA [88]	*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%**

Table 4. Task 1: Subjects were shown ground truth and generated image and asked for preference. Task 2: Subjects had to decide between two generated images. More details in E.3.6

Experiments

Method	FID ↓	IS ↑	PSNR ↑	SSIM ↑	N_{params}	$[\frac{\text{samples}}{s}](*)$
Image Regression [72]	15.2	121.1	**27.9**	**0.801**	625M	N/A
SR3 [72]	5.2	**180.1**	26.4	0.762	625M	N/A
LDM-4 (ours, 100 steps)	2.8†/4.8‡	166.3	$24.4_{\pm 3.8}$	$0.69_{\pm 0.14}$	**169M**	4.62
emphLDM-4 (ours, big, 100 steps)	**2.4**†/**4.3**‡	174.9	$24.7_{\pm 4.1}$	$0.71_{\pm 0.15}$	552M	4.5
LDM-4 (ours, 50 steps, guiding)	4.4†/6.4‡	153.7	$25.8_{\pm 3.7}$	$0.74_{\pm 0.12}$	184M	0.38

Table 5. $\times 4$ upscaling results on ImageNet-Val. (256^2); †: FID features computed on validation split, ‡: FID features computed on train split; *: Assessed on a NVIDIA A100

Model (*reg.*-type)	train throughput samples/sec.	sampling throughput† @256	@512	train+val hours/epoch	FID@2k epoch 6
LDM-1 (no first stage)	0.11	0.26	0.07	20.66	24.74
LDM-4 (*KL*, w/ attn)	0.32	0.97	0.34	7.66	15.21
LDM-4 (*VQ*, w/ attn)	0.33	0.97	0.34	7.04	14.99
LDM-4 (*VQ*, w/o attn)	0.35	0.99	0.36	6.66	15.95

Table 6. Assessing inpainting efficiency. †: Deviations from Fig. 7 due to varying GPU settings/batch sizes *cf*. the supplement.

Experiments

Figure 11. Qualitative results on object removal with our *big, w/ ft* inpainting model. For more results, see Fig. 22.

Experiments

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 [88]†	12.31	**0.243**± 0.038	2.23	**0.134**± 0.080
LaMa [88]	12.0	**0.24**	2.21	0.14
CoModGAN [107]	10.4	0.26	1.82	0.15
RegionWise [52]	21.3	0.27	4.75	0.15
DeepFill v2 [104]	22.1	0.28	5.20	0.16
EdgeConnect [58]	30.5	0.28	8.37	0.16

Table 7. Comparison of inpainting performance on 30k crops of size 512×512 from test images of Places [108]. The column *40-50%* reports metrics computed over hard examples where 40-50% of the image region have to be inpainted. †recomputed on our test set, since the original test set used in [88] was not available.

Limitation

- **Limitations**

  While LDMs significantly reduce computational requirements compared to pixel-based approaches, their sequential sampling process is still slower than that of GANs.

  Moreover, the use of LDMs can be questionable when high precision is required: although the loss of image quality is very small in our f=4 autoencoding models, their reconstruction capability can become a bottleneck for tasks that require fine-grained accuracy in pixel space.

- **Societal Impact**

  Generative models for media like imagery are a double-edged sword: On the one hand, they enable various creative applications, and in particular approaches like ours that **reduce the cost of training** and inference have the potential to facilitate access to this technology and **democratize** its exploration.

  On the other hand, it also means that it becomes easier to create and disseminate manipulated data or spread misinformation and spam. In particular, the deliberate manipulation of images ("**deep fakes**") is a common problem in this context, and women in particular are disproportionately affected by it.

  Generative models can also reveal their training data, which is of great concern when the data contain sensitive or personal information and were collected without explicit consent.

Conclusion

- We have presented latent diffusion models, a simple and efficient way to significantly improve both the training and sampling efficiency of denoising diffusion models without degrading their quality.

- Based on this and our cross-attention conditioning mechanism, our experiments could demonstrate favorable results compared to state-of-the-art methods across a wide range of conditional image synthesis tasks without task-specific architectures.

Appendix

G. Details on Autoencoder Models

We train all our autoencoder models in an adversarial manner following [23], such that a patch-based discriminator D_ψ is optimized to differentiate original images from reconstructions $\mathcal{D}(\mathcal{E}(x))$. To avoid arbitrarily scaled latent spaces, we regularize the latent z to be zero centered and obtain small variance by introducing an regularizing loss term L_{reg}.
We investigate two different regularization methods: (i) a low-weighted Kullback-Leibler-term between $q_\mathcal{E}(z|x) = \mathcal{N}(z; \mathcal{E}_\mu, \mathcal{E}_{\sigma^2})$ and a standard normal distribution $\mathcal{N}(z; 0, 1)$ as in a standard variational autoencoder [46, 69], and, (ii) regularizing the latent space with a vector quantization layer by learning a codebook of $|\mathcal{Z}|$ different exemplars [96].
To obtain high-fidelity reconstructions we only use a very small regularization for both scenarios, *i.e.* we either weight the $\mathbb{KL}$ term by a factor $\sim 10^{-6}$ or choose a high codebook dimensionality $|\mathcal{Z}|$.

The full objective to train the autoencoding model $(\mathcal{E}, \mathcal{D})$ reads:

$$L_{\text{Autoencoder}} = \min_{\mathcal{E},\mathcal{D}} \max_{\psi} \Big(L_{rec}(x, \mathcal{D}(\mathcal{E}(x))) - L_{adv}(\mathcal{D}(\mathcal{E}(x))) + \log D_\psi(x) + L_{reg}(x; \mathcal{E}, \mathcal{D}) \Big) \tag{25}$$

DM Training in Latent Space Note that for training diffusion models on the learned latent space, we again distinguish two cases when learning $p(z)$ or $p(z|y)$ (Sec. 4.3): (i) For a KL-regularized latent space, we sample $z = \mathcal{E}_\mu(x) + \mathcal{E}_\sigma(x) \cdot \varepsilon =: \mathcal{E}(x)$, where $\varepsilon \sim \mathcal{N}(0, 1)$. When rescaling the latent, we estimate the component-wise variance

$$\hat{\sigma}^2 = \frac{1}{bchw} \sum_{b,c,h,w} (z^{b,c,h,w} - \hat{\mu})^2$$

from the first batch in the data, where $\hat{\mu} = \frac{1}{bchw} \sum_{b,c,h,w} z^{b,c,h,w}$. The output of $\mathcal{E}$ is scaled such that the rescaled latent has unit standard deviation, *i.e.* $z \leftarrow \frac{z}{\hat{\sigma}} = \frac{\mathcal{E}(x)}{\hat{\sigma}}$. (ii) For a VQ-regularized latent space, we extract z *before* the quantization layer and absorb the quantization operation into the decoder, *i.e.* it can be interpreted as the first layer of $\mathcal{D}$.

Appendix

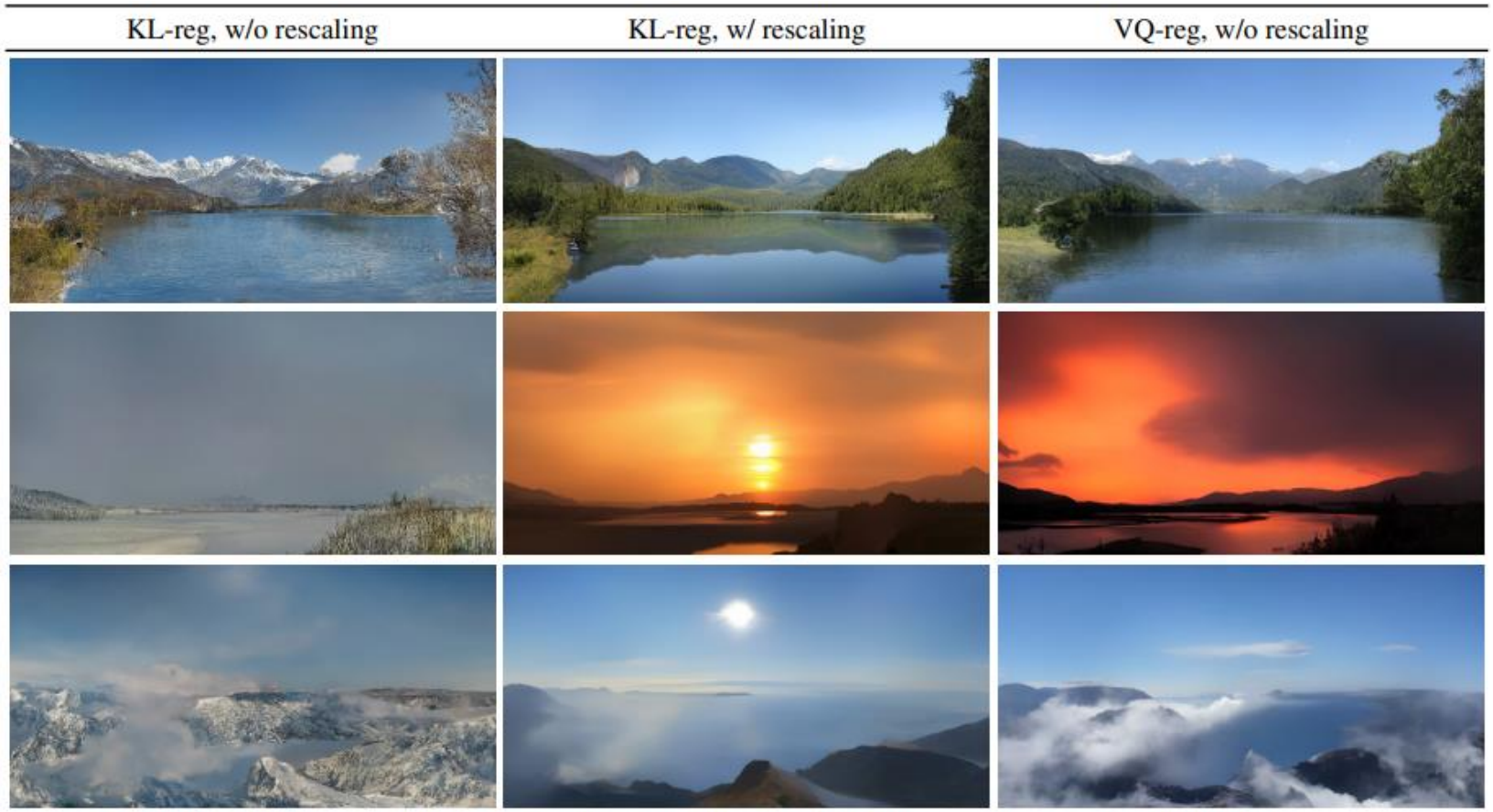


Figure 15. Illustrating the effect of latent space rescaling on convolutional sampling, here for semantic image synthesis on landscapes. See Sec. 4.3.2 and Sec. D.1.

As discussed in Sec. 4.3.2, the signal-to-noise ratio induced by the variance of the latent space (*i.e.* $\mathrm{Var}(z)/\sigma_t^2$) significantly affects the results for convolutional sampling. For example, when training a LDM directly in the latent space of a KL-regularized model (see Tab. 8), this ratio is very high, such that the model allocates a lot of semantic detail early on in the reverse denoising process. In contrast, when rescaling the latent space by the component-wise standard deviation of the latents as described in Sec. G, the SNR is descreased. We illustrate the effect on convolutional sampling for semantic image synthesis in Fig. 15. Note that the VQ-regularized space has a variance close to 1, such that it does not have to be rescaled.

Reference

- https://arxiv.org/abs/2208.05140
- CLIP: http://dmqm.korea.ac.kr/activity/seminar/365
- Score-based Generative Models and Diffusion Models : http://dmqm.korea.ac.kr/activity/seminar/352
- Multimodal Learning: http://dmqm.korea.ac.kr/activity/seminar/341
- Multimodal Representation Learning :
- How to narrow heterogeneity gap: http://dmqm.korea.ac.kr/activity/seminar/373
- Denoising Diffusion Probabilistic Models: https://www.youtube.com/watch?v=_JQSMhqXw-4
- DiffusionCLIP: https://www.youtube.com/watch?v=ZKAXgg_OAE0
- VATT: https://www.youtube.com/watch?v=rgXxAFIBido
- ImageBART: https://www.youtube.com/watch?v=uy3MEwBTR4o
- NÜWA: https://www.youtube.com/watch?v=jDlLnNiRXTc
- data2vec: https://www.youtube.com/watch?v=cFPCKCglpuE
- DALL-E 1: https://www.youtube.com/watch?v=az-OV47oKvA
- DALL-E 2: https://www.youtube.com/watch?v=WRkClaOAjkA
- Retrieval-Augmented Diffusion Models: https://www.youtube.com/watch?v=Ktgt7bcXLYI
- Structured Denoising Diffusion Models in Discrete State-Spaces: https://www.youtube.com/watch?v=kZz4HLLS-q0
- Score-based Generative Modeling Through Stochastic Differential Equations: https://www.youtube.com/watch?v=uG2ceFnUeQU
- Denoising Diffusion Probabilistic Models: https://www.youtube.com/watch?v=1j0W_lu55nc&t=24s
- GLIDE: https://www.youtube.com/watch?v=ajBRo40-Kjs, https://www.youtube.com/watch?v=Ntjm2Icrh0E
- Diffusion Model 설명: https://www.youtube.com/watch?v=uFoGaIVHfoE
- Stable Diffusion: https://www.youtube.com/watch?v=ibB6aR9xA1Q
- DDPM 소개 블로그 (유명): https://angusturner.github.io/generative_models/2021/06/29/diffusion-probabilistic-models-I.html
- DDPM + 총망라 블로그 (유명): https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- 유재준 교수님: https://www.youtube.com/watch?v=m0sehjymZNU
- 조한샘: https://www.youtube.com/watch?v=ix09cEMmwb0

Thank you for your Attention….!