Dive into Diffusion Model:
DDPM to DDPO

YAI 생성논문팀 분리세션

24.09.11 / 14기 김민규

CONTENTS

0.

Intro

0. Intro

Our Team

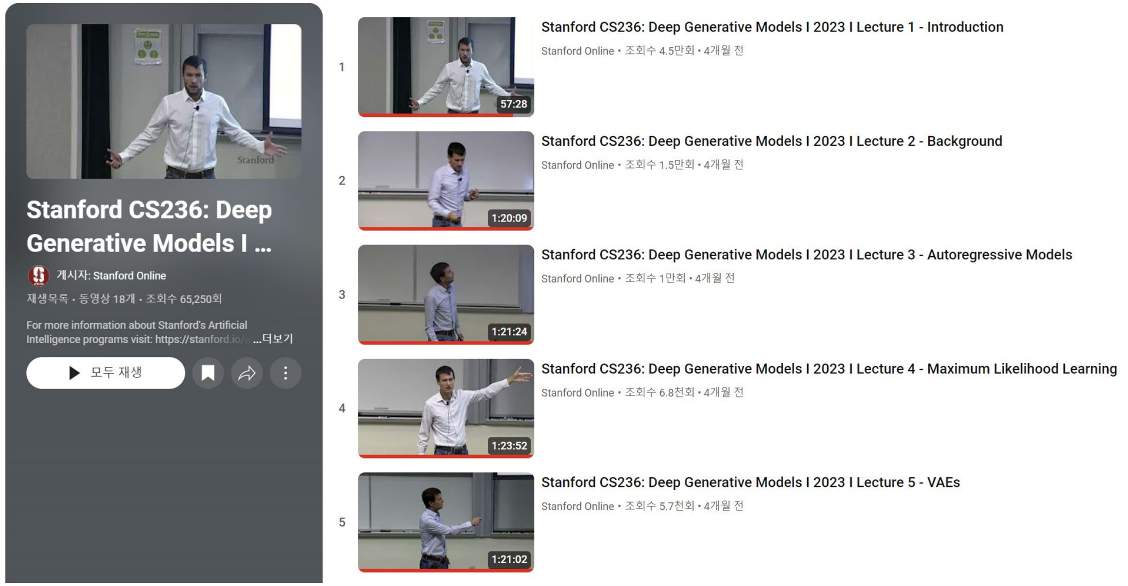
생성 논문

활동기간	14기 가을전반기
팀 설명	생성모델의 세계를 논문과 강의로 탐험하며, AI의 미래를 한 걸음씩 그려나갑니다.
Members	(공과대학 건설환경공학) 고현아, 김민규, 윤찬용, 김성은, 이상훈, (공과대학 신소재공학) 양준호

댓글 추가

커리큘럼

주차	일자	내용	비고	발제자
1주차	24.09.02	CS236 Week 2~6 + GAN	https://arxiv.org/abs/1406.2661	고현아
2주차	24.09.10	DDPM	https://arxiv.org/abs/2006.11239	이상훈



Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio‡

0. Intro

Our Team

생성 논문

활동기간	14기 가을전반기
팀 설명	비어 있음
Members	(공과대학 건설환경공학) 고현아, 김민규, 윤찬용, 김성은, 이상훈, (공과대학 신소재공학) 양준호

나는 지금 Yonsei Artificial Intelligence (YAI)라는 학회에서 활동 중이고, 이번 가을 전반기 세션은 생성 논문 팀에서 활동하게 됐어. 우리 팀의 활동 내용은 아래와 같아.

- 생성모델 관련 기초부터 심화 논문을 읽고, 이를 매주마다 돌아가며 발제한다.
- Stanford CS236 강의를 보며 이를 리뷰한다.

팀 설명이 아직 정해지지 않아서 한 문장으로 간단하게 채워볼까 하는데, 어떻게 적으면 재치있으면서도 우리의 활동이 잘 드러나게 적을 수 있을지 알려줘!

메모리 업데이트됨

팀 설명에 재치 있으면서도 활동 내용을 잘 담으려면, 이런 문장을 고려해볼 수 있을 것 같아:

"생성모델의 세계를 논문과 강의로 탐험하며, AI의 미래를 한 걸음씩 그려나갑니다."

이 문장은 너희 팀이 논문과 강의를 통해 생성모델을 깊이 있게 탐구하는 활동을 강조하면서도, 탐험하듯 배운다는 재미있는 느낌을 줄 수 있어!

음 좀 구려 보이지만 일단 해볼게. 고마워!

네, 마음에 들지 않더라도 새로운 시도를 함께 해보는 것도 좋은 경험이니까요! 혹시 다른 아이디어가 필요하면 언제든 알려줘. 더 세련된 버전을 고민해볼 수 있어!

0. Intro

Generative Model

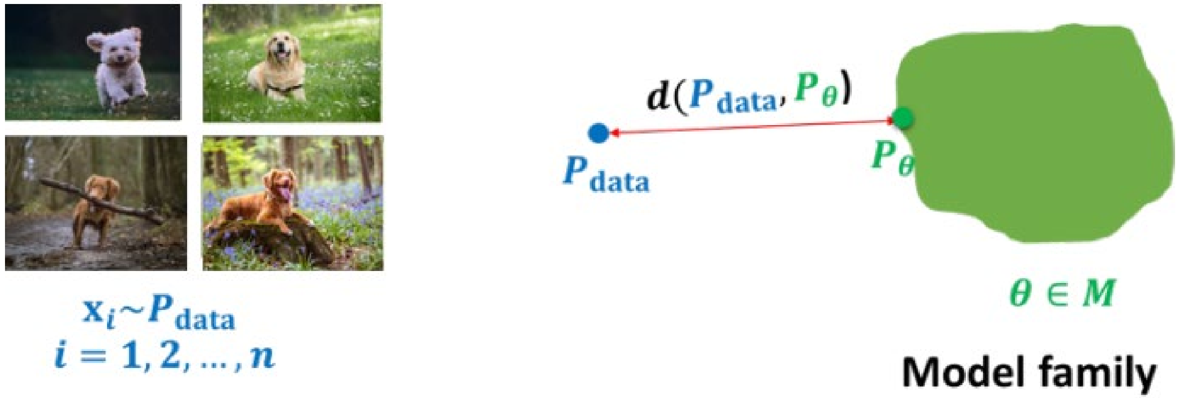


Figure adapted from Stefano Ermon, CS236 (Deep Generative Model), 2023.

1) **Generation**: If we sample $x_{new} \sim p(x)$, this new data should look like original one.

2) **Density Estimation**: $p(x)$ should be high only for true x. (Outlier Detection)

3) **Unsupervised Representation Learning**: Be able to learn the data's structure.

Generative Model

Taxonomy of Generative Models

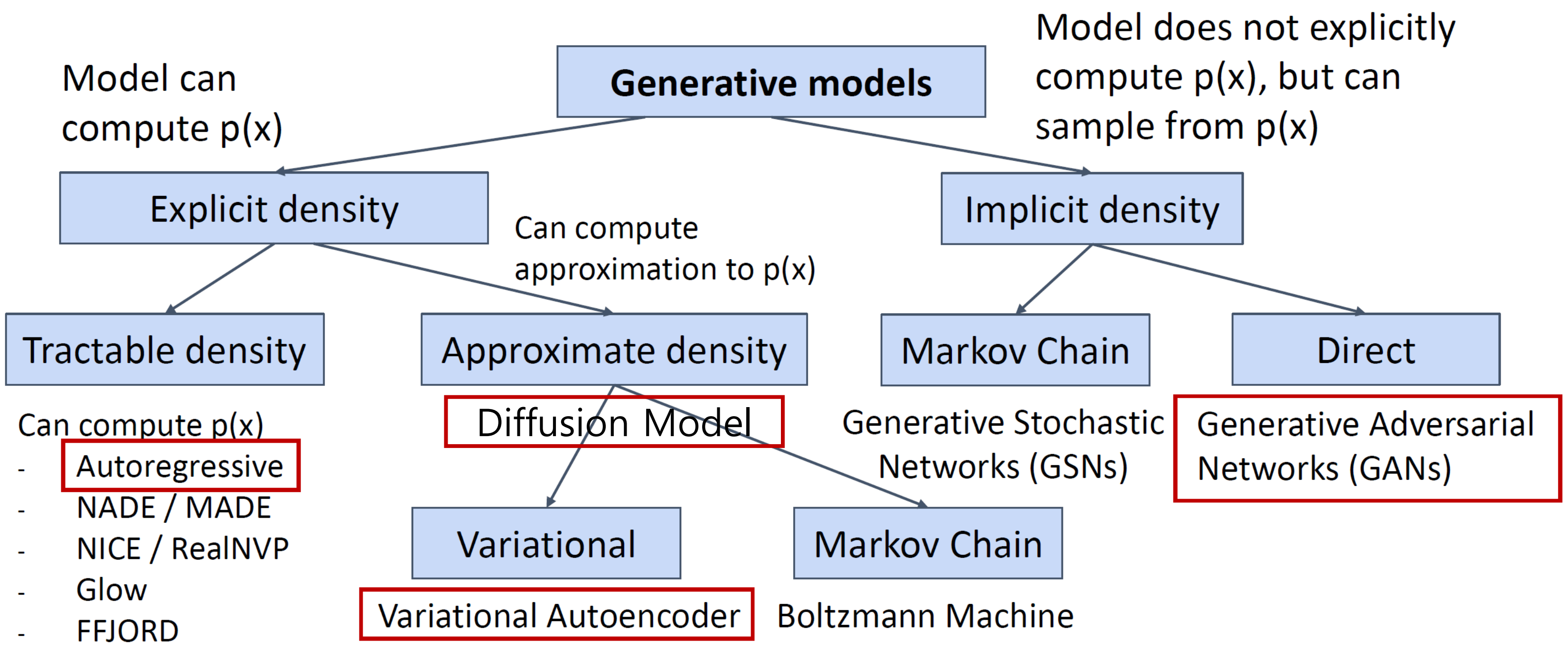
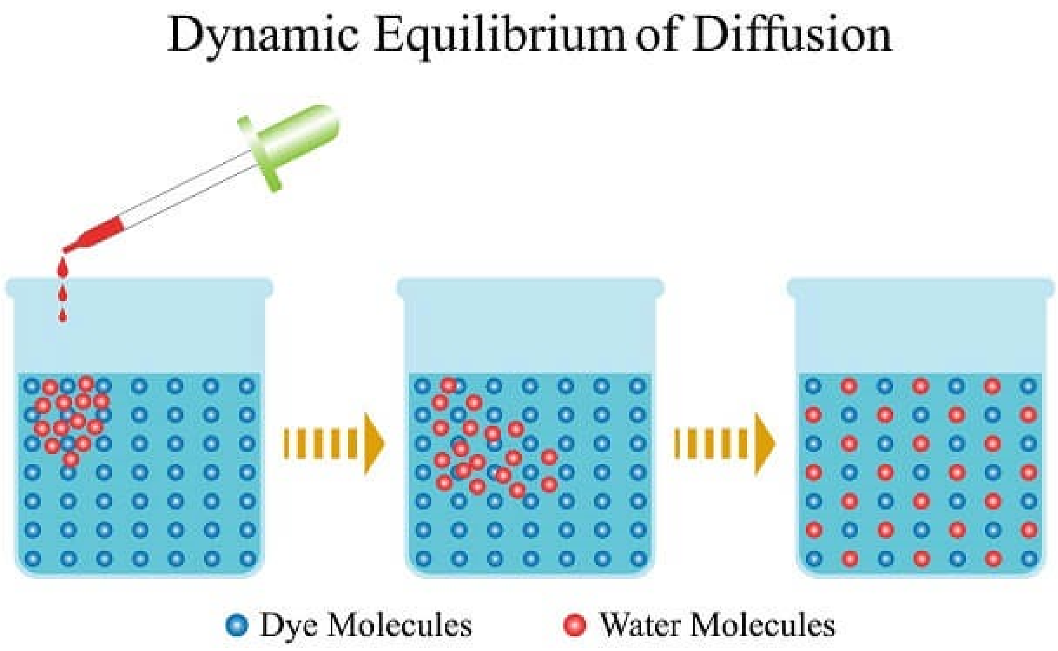


Figure adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

0. Intro

Diffusion Model

Motivation

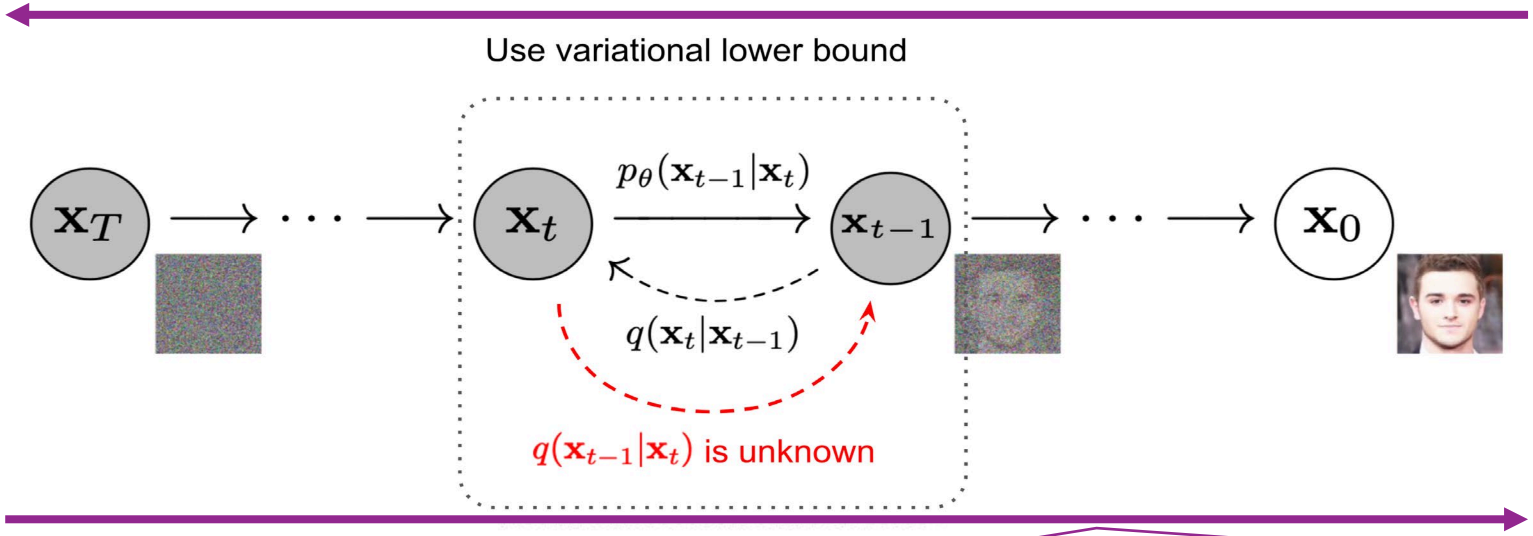


0. Intro

Diffusion Model

Overall Architecture

Forward Diffusion Process = Encoder in VAE
(By adding Gaussian Random Noise)



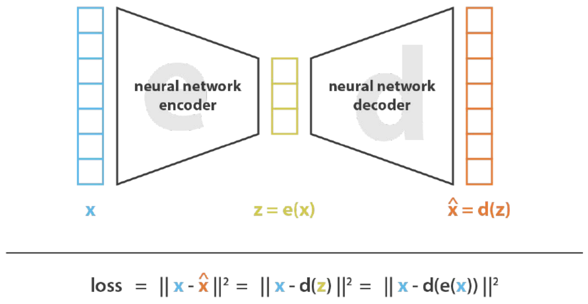
Reverse Diffusion Process (Denoising) = Decoder in VAE
(Goal of the Diffusion Model = Denoising DPM (DDPM))

Diffusion Model

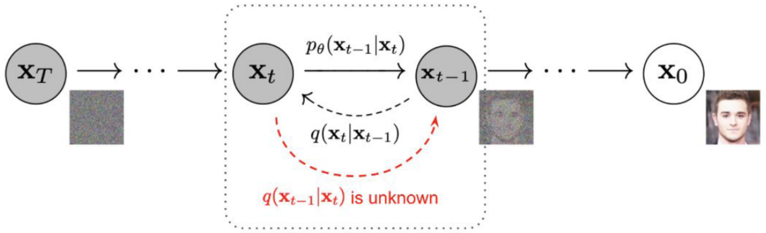
Q) Why Diffusion Model Works?

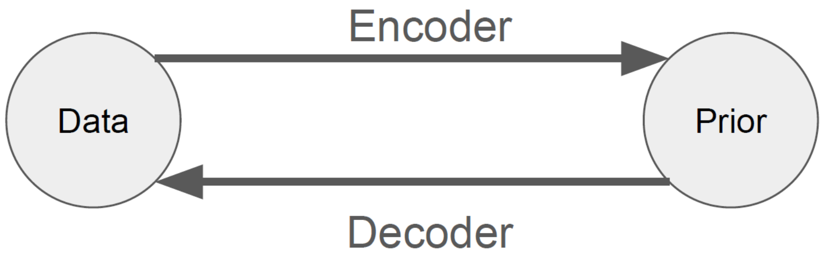
A) Vanilla VAE has single latent variable, while **Diffusion Model has 1000 to 4000~Inf latent variables!**

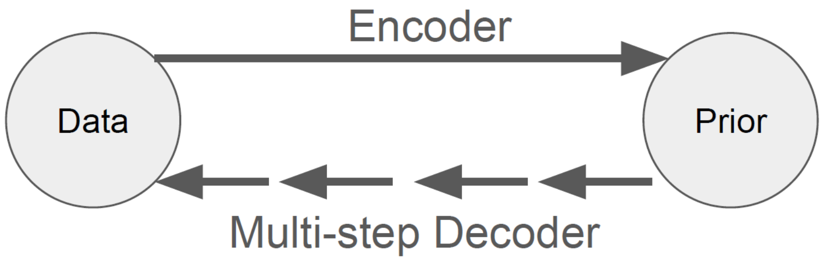
< VAE >



< Diffusion Model >







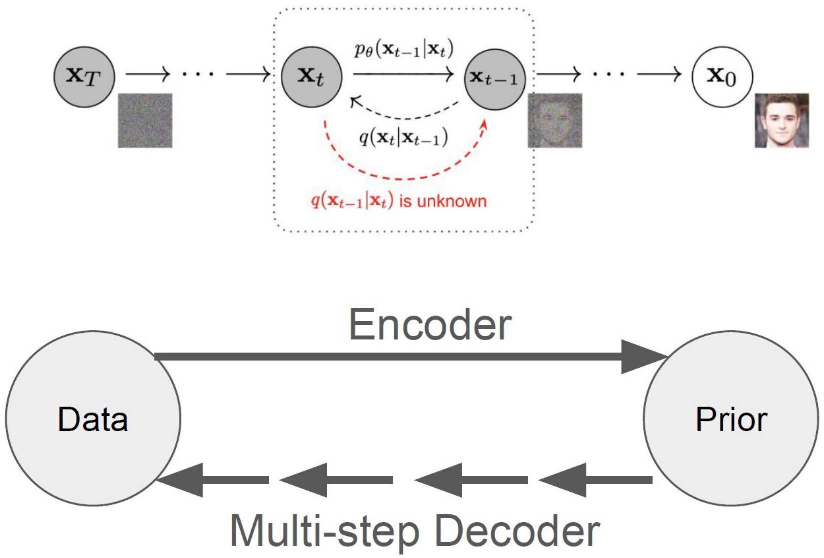
0. Intro

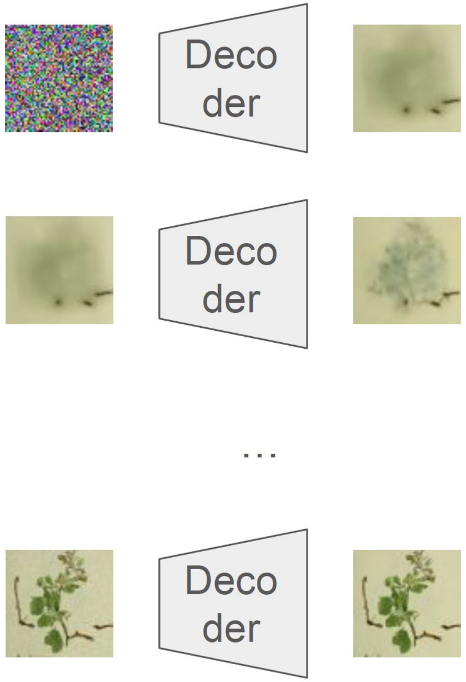
Diffusion Model

Q) Why Diffusion Model Works?

A) Vanilla VAE has single latent variable, while **Diffusion Model has 1000 to 4000~Inf latent variables!**

< Diffusion Model >





1.

DDPM

1. DDPM

Forward Process

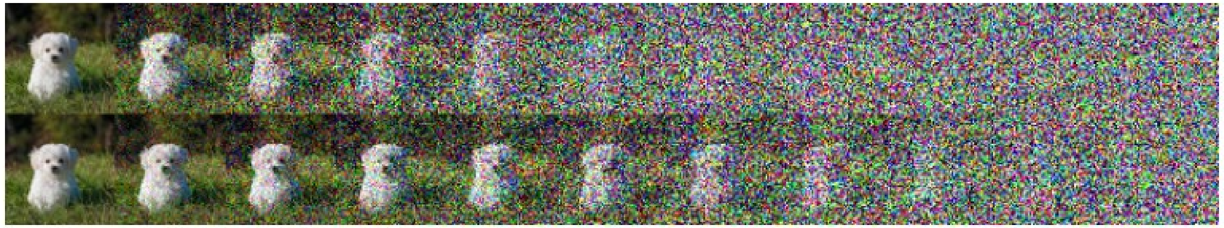
= Adding small amount of Gaussian noise

vs. VAE: Not having parameter in encoder

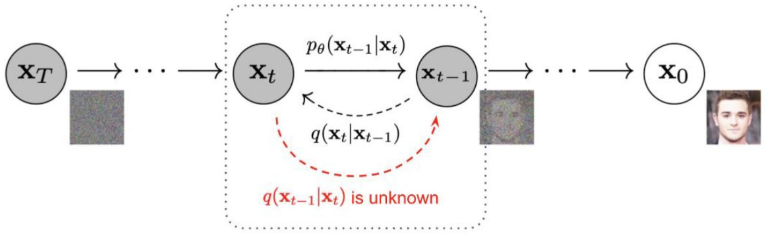
(pre-defined Noise Generator = Hyperparameter!)

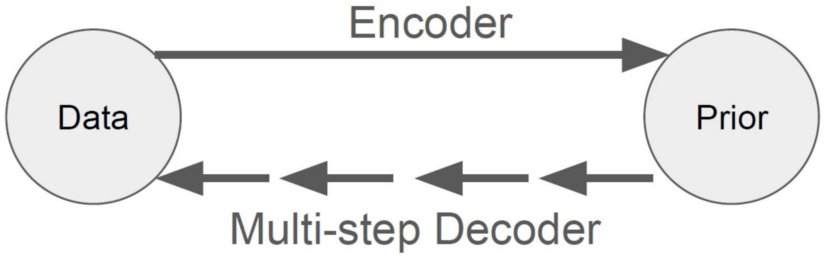
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



< Diffusion Model >





1. DDPM

Forward Process

= Adding small amount of Gaussian noise

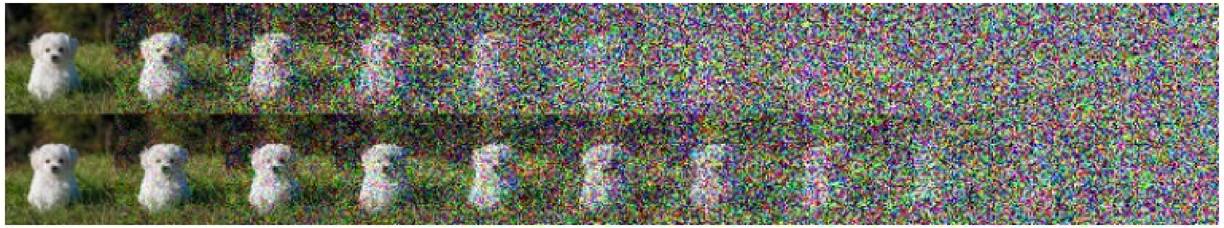
to the sample in T steps, producing a sequence

of Noisy sampes x_1 to x_T. (able in 1-step!)

※ β_t: (scaling down) variance from previous image

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



$$\begin{aligned} x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon, where\ \epsilon \sim N(0, I) \\ &= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon\right) + \sqrt{1-\alpha_t} \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \left(\sqrt{\alpha_t(1-\alpha_{t-1})}\epsilon + \sqrt{1-\alpha_t}\epsilon\right) \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon \\ &\ldots. \\ &\Rightarrow x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \end{aligned}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad \text{where, } \alpha_t := 1-\beta_t,\ \bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s$$

$$q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_T; \sqrt{\bar{\alpha}_T}\mathbf{x}_0, (1-\bar{\alpha}_T)\mathbf{I}) \approx \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \mathbf{I})$$

1. DDPM

Reverse Process

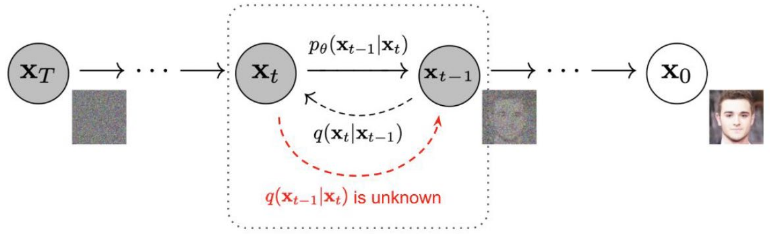
Forward = Gaussian // Reverse = ???

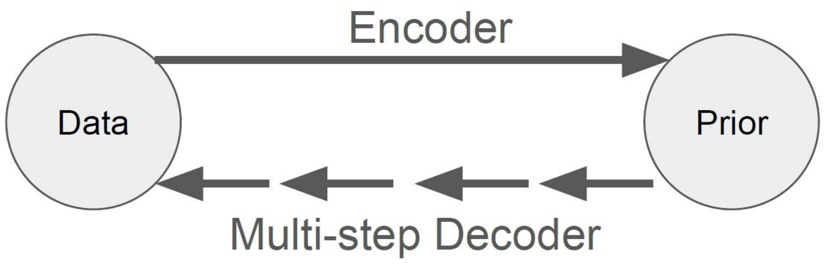
In 1949, Feller showed that the reverse will also be Gaussian for very small variance β > 0.

Then, how can we estimate it?

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

< Diffusion Model >





1. DDPM

Reverse Process

$$
\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t}+\frac{(\mathbf{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}}-\frac{(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_t)^2}{1-\bar{\alpha}_t}\right)\right) \\
&= \exp\left(-\frac{1}{2}\left(\frac{\mathbf{x}_t^2-2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1}+\alpha_t\mathbf{x}_{t-1}^2}{\beta_t}+\frac{\mathbf{x}_{t-1}^2-2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1}+\bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}}-\frac{(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_t)^2}{1-\bar{\alpha}_t}\right)\right) \\
&= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^2-\left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\mathbf{x}_{t-1}+C(\mathbf{x}_t,\mathbf{x}_0)\right)\right)
\end{aligned}
$$

$$
\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right) = 1/\left(\frac{\alpha_t-\bar{\alpha}_t+\beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}\right) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t
$$

$$
\begin{aligned}
\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) &= \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)/\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right) \\
&= \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\cdot\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0
\end{aligned}
$$

Reverse Process

(*) Recall that when we merge two Gaussians with different variance, $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$, the new distribution is $\mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$. Here the merged standard deviation is $\sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$.

Thanks to the nice property, we can represent $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t)$ and plug it into the above equation and obtain:

$$\tilde{\boldsymbol{\mu}}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t)$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$$

$\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$:

1. DDPM

Reverse Process

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T}+\sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

Thanks to the nice property, we can represent $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t)$ and plug it into the above equation and obtain:

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t) \\ &= \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\Big)\end{aligned}$$

$$L_t = D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \le t \le T-1$$

KL Divergence between
two Gaussian Distribution

1. DDPM

Reverse Process

The loss term L_t is parameterized to minimize the difference from $\tilde{\boldsymbol{\mu}}$:

Make NN of θ estimating noise!
Input = initial noise x_0 + time step t

$$
\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\epsilon}\Big[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t,t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0)-\boldsymbol{\mu}_\theta(\mathbf{x}_t,t)\|^2\Big] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\Big[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2}\Big\|\frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\Big)-\frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\Big)\Big\|^2\Big] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\Big[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)\|^2\Big] \\
&= \mathbb{E}_{\mathbf{x}_0,\epsilon}\Big[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t,t)\|^2\Big]
\end{aligned}
$$

Algorithm 1 Training

repeat
- $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
- $t \sim \mathrm{Uniform}(\{1,\ldots,T\})$
- $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
- Take gradient descent step on
  $\nabla_\theta \left\|\epsilon-\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\epsilon,t)\right\|^2$

until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
for $t=T,\ldots,1$ **do**
- $\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ if $t>1$, else $\mathbf{z}=\mathbf{0}$
- $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t,t)\right)+\sigma_t\mathbf{z}$

end for
return $\mathbf{x}_0$

1. DDPM

Results

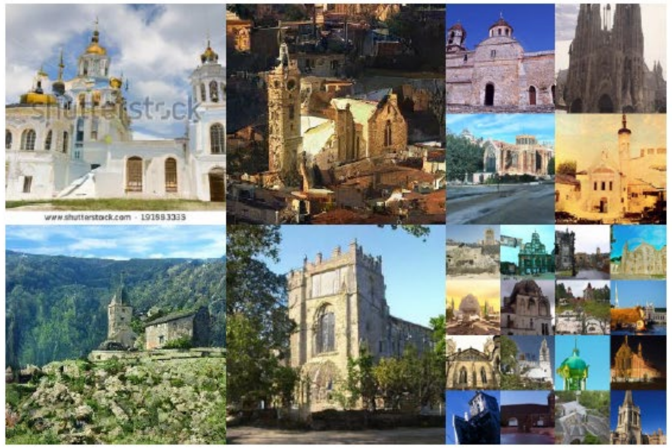


Figure 3: LSUN Church samples. FID=7.89

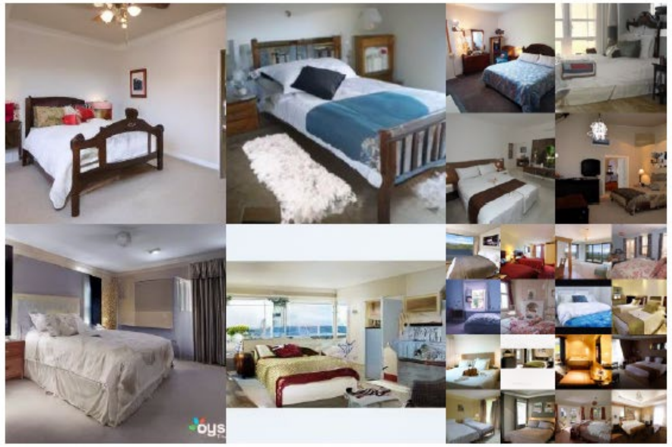


Figure 4: LSUN Bedroom samples. FID=4.90

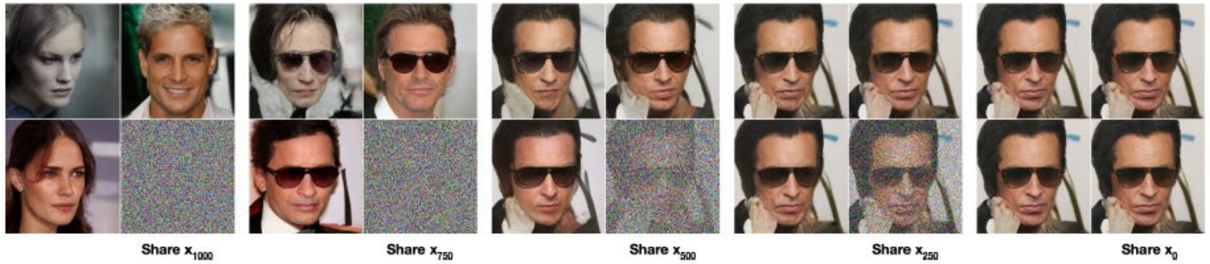


Figure 7: When conditioned on the same latent, CelebA-HQ 256 × 256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

2.

LDM

Motivation

DDPM works well compared with VAE!

Con) The dimension does not change.

= Computationally inefficient

= Inflexible Generation

Stable Diffusion arises here! (by LDM)

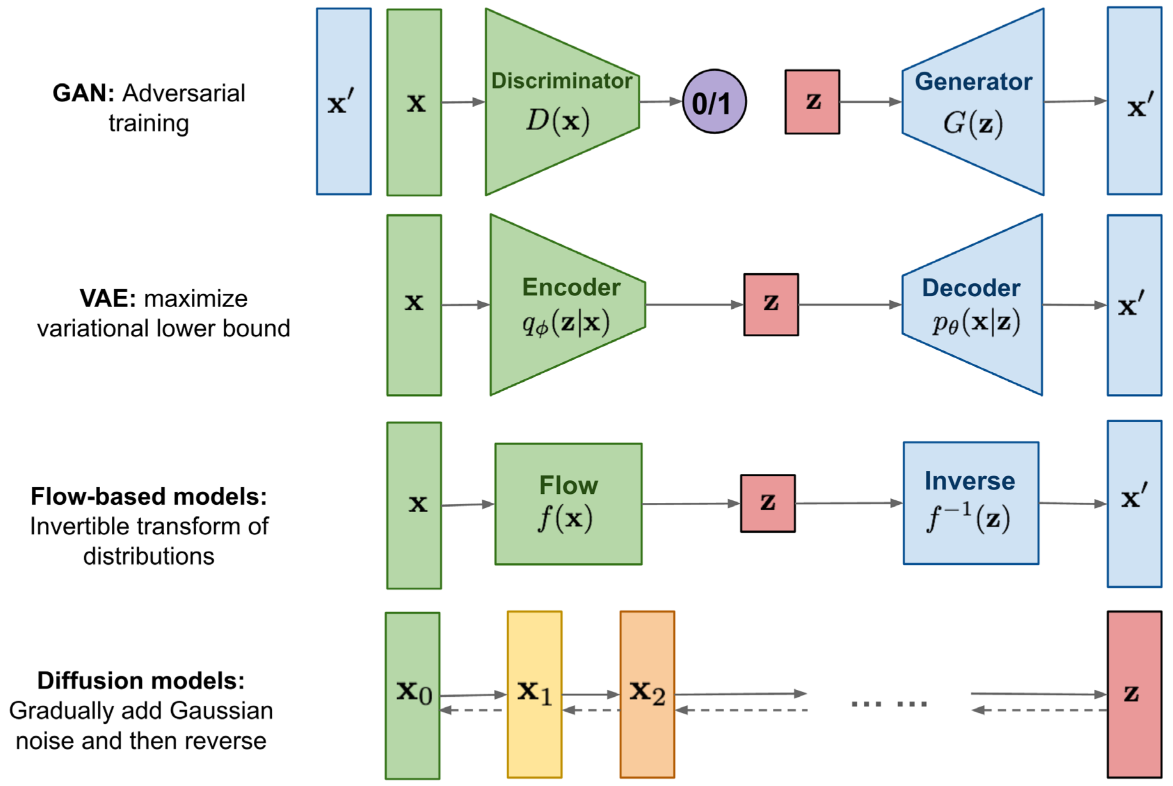
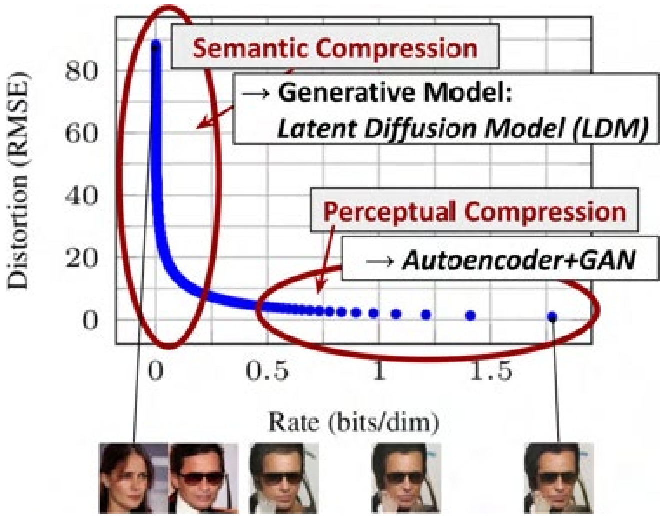


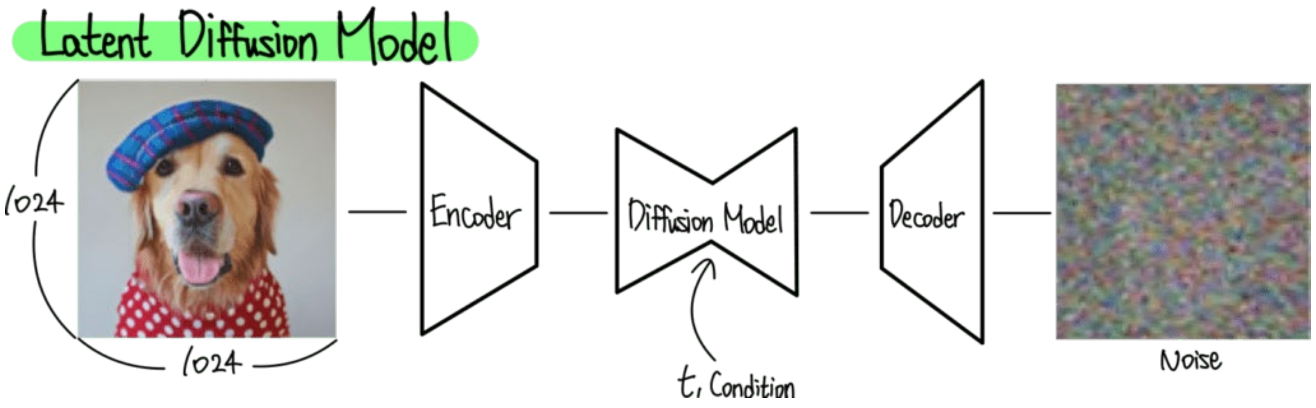
Figure adapted from Weng et al., What are Diffusion Models?, 2021.

Motivation

Q. How can we reduce the training and inference cost?

A. Conduct main task of feature extracting (= Semantic Compression) with lowered dimension,
and do remaining task (= Perceptual Compression) using Autoencoder while lowering dimension!





2. LDM

Architecture

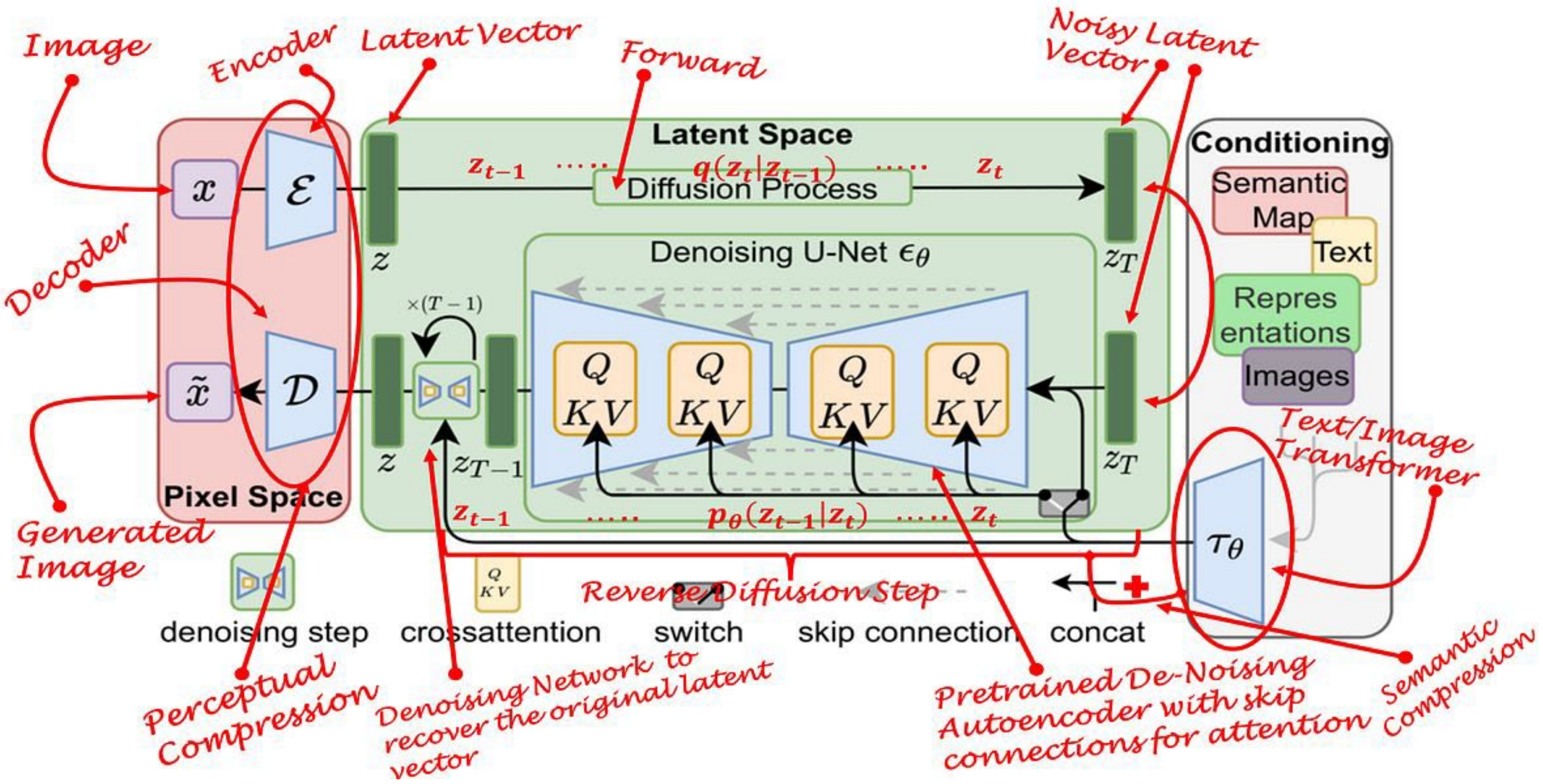


Figure adapted from Weng et al., What are Diffusion Models?, 2021.

2. LDM

Results

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

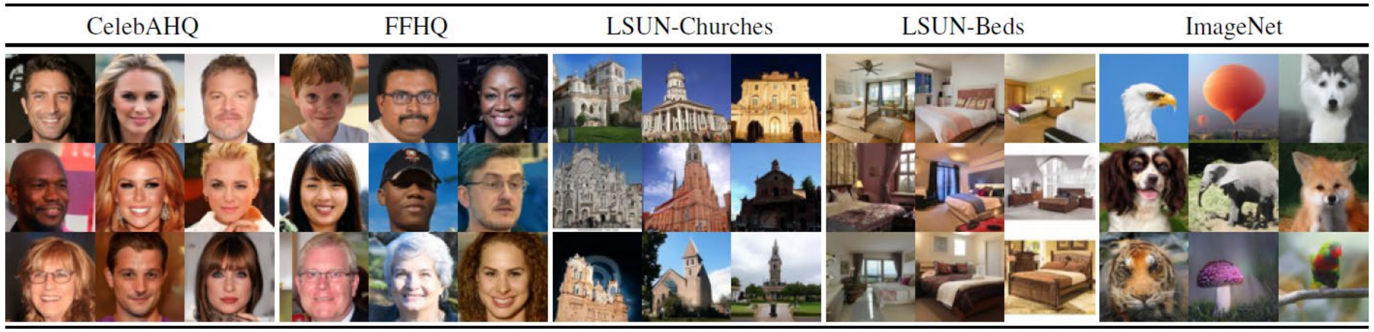
FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
ImageBART [21]	9.57	-	-
U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
UDM [43]	5.54	-	-
StyleGAN [41]	4.16	0.71	0.46
ProjectedGAN [76]	**3.08**	0.65	0.46
LDM-4 (ours, 200-s)	4.98	**0.73**	**0.50**

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

Text-Conditional Image Synthesis				
Method	FID ↓	IS↑	N_{params}	
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$0.33	1.45B	250 DDIM steps
*LDM-KL-8-G**	12.63	**30.29**$\pm$**0.42**	1.45B	250 DDIM steps, c.f.g. [32] $s = 1.5$

Results


CelebAHQ
FFHQ
LSUN-Churches
LSUN-Beds
ImageNet


Text-to-Image Synthesis on LAION. 1.45B Model.
'A street sign that reads "Latent Diffusion" '
'A zombie in the style of Picasso'
'An image of an animal half mouse half octopus'
'An illustration of a slightly conscious neural network'
'A painting of a squirrel eating a burger'
'A watercolor painting of a chair that looks like an octopus'
'A shirt with the inscription: "I love generative models!" '
LATENT DIFFUSION
LATETEN DIFFUSION
I Generative Models!

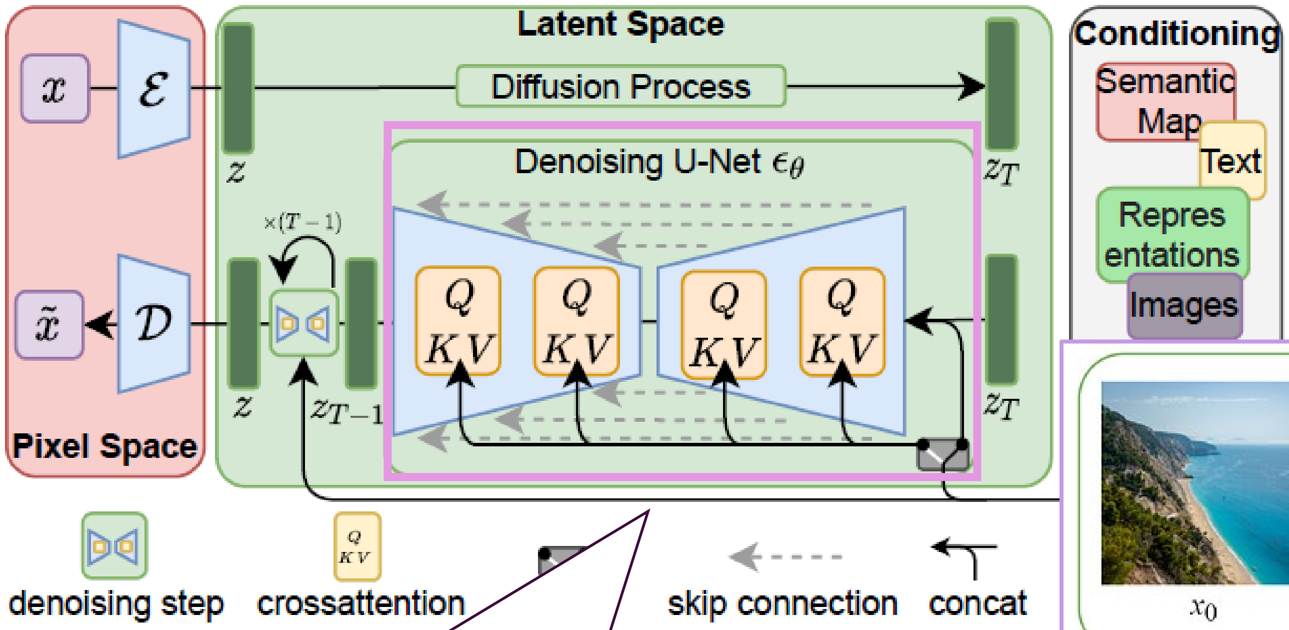

input
result

3.

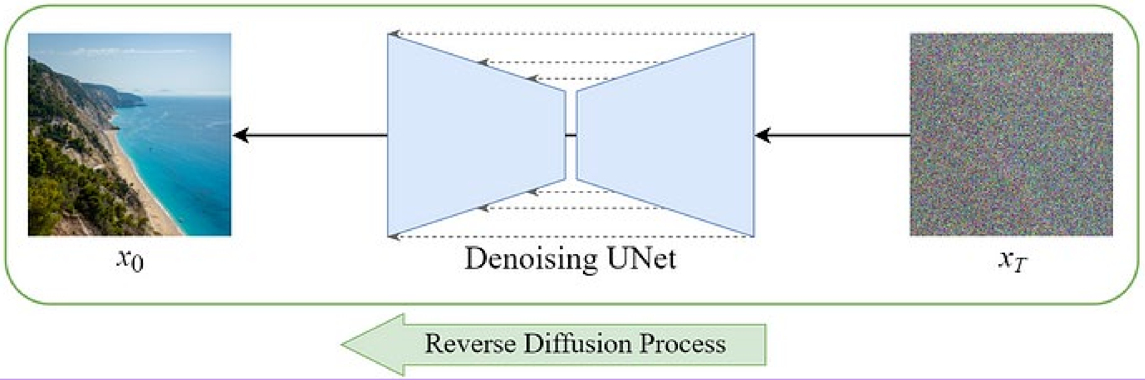
DiT

3. DiT

LDM Revisiting



U-Net as a backbone..?



LDM Revisiting

DDPM uses U-Net backbone for reverse diffusion process, and so does LDM.

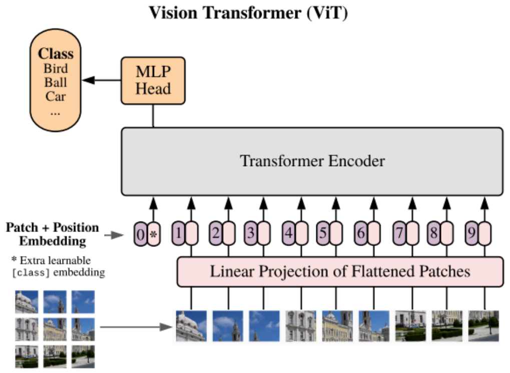
However, U-Net's inductive bias is not crucial to the performance of Diffusion Model.

Diffusion Transformer (DiT)

= Adapts Vision Transfomer (ViT) architecture that operates on latent patches

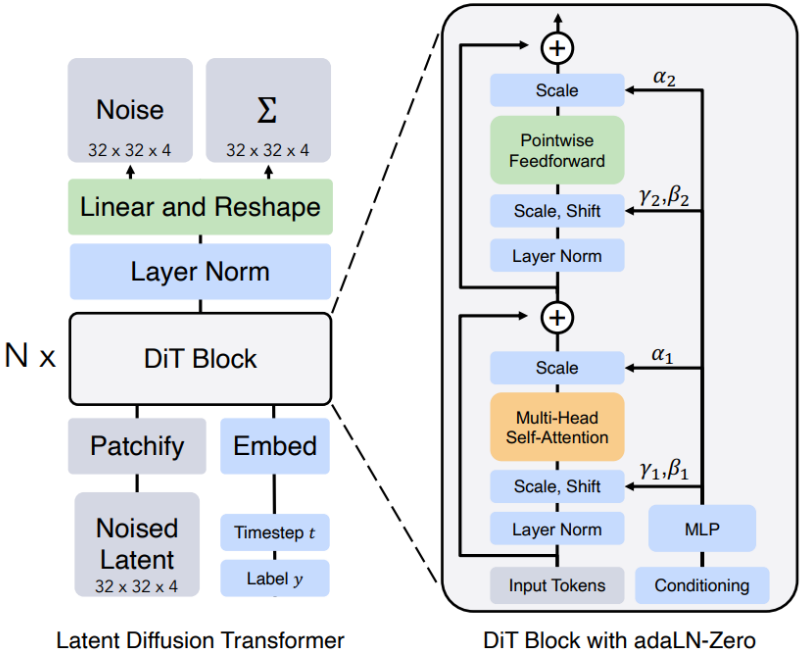
= Can inherit best practices and training methods from other domains

= Retains scalability, robustness, and efficiency



3. DiT

Architecture



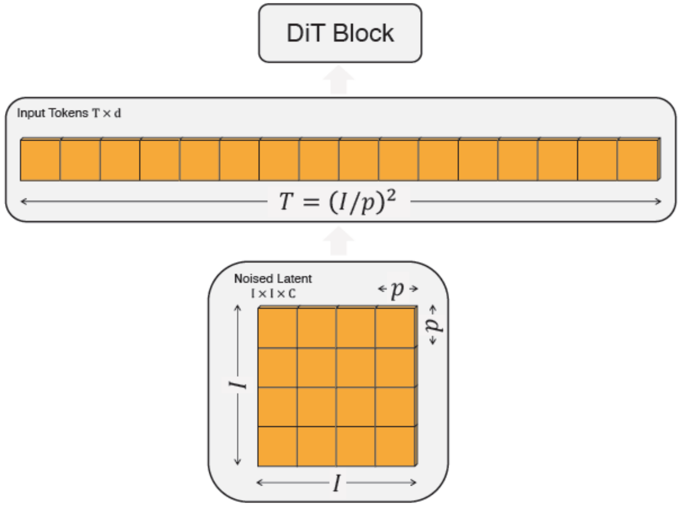
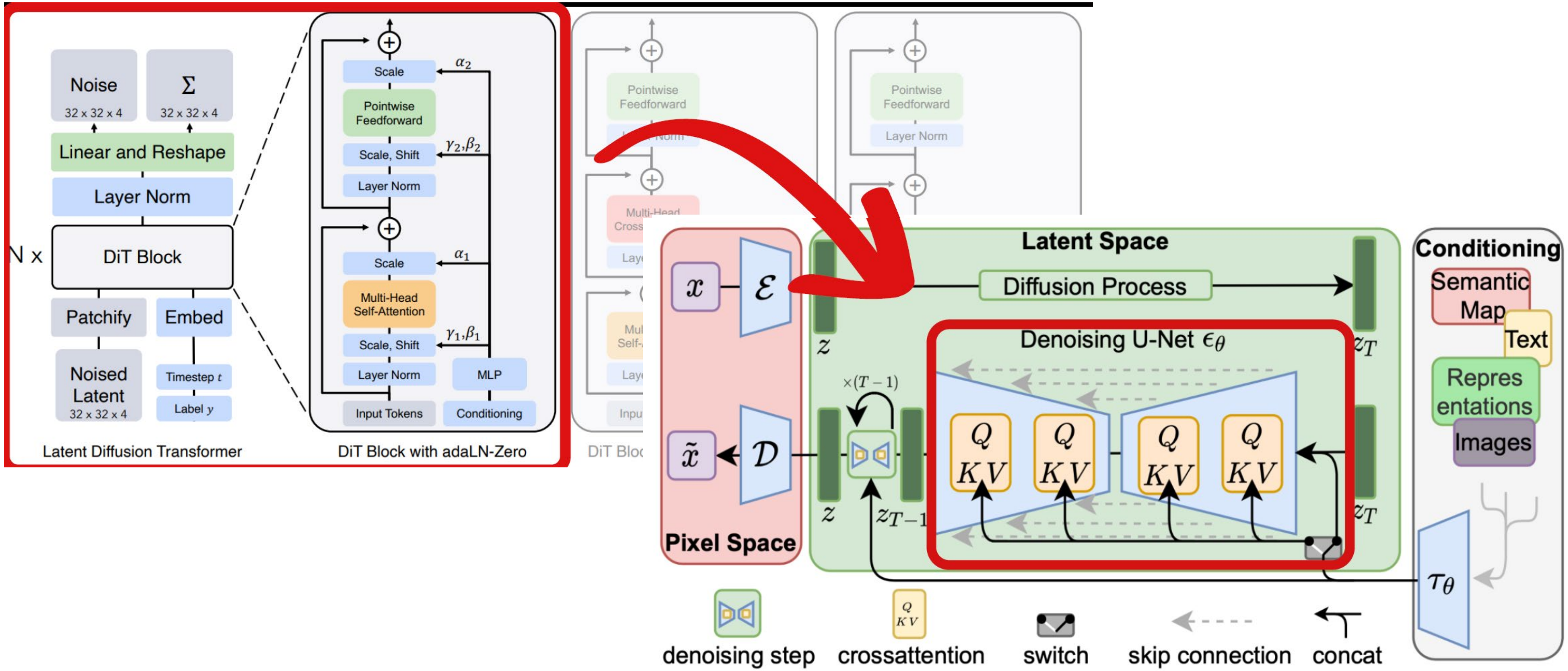


Figure 4. **Input specifications for DiT.** Given patch size $p \times p$, a spatial representation (the noised latent from the VAE) of shape $I \times I \times C$ is "patchified" into a sequence of length $T = (I/p)^2$ with hidden dimension d. A smaller patch size p results in a longer sequence length and thus more Gflops.

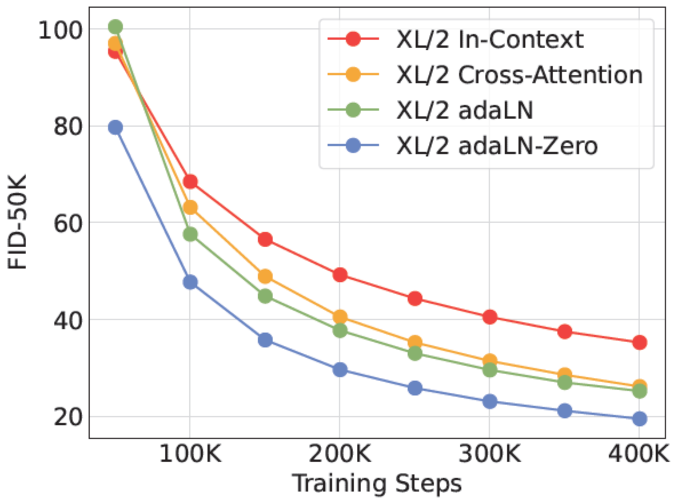
3. DiT

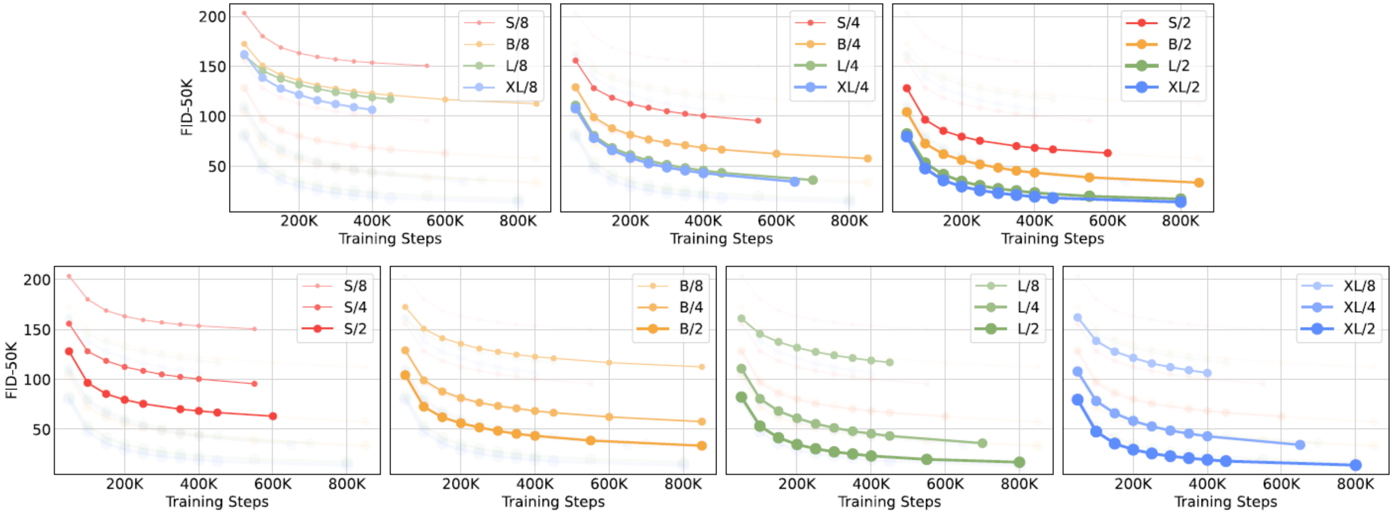
Architecture



Results

※ FID (Frechet Inception Distance): Metric for feature distance between real & generated images



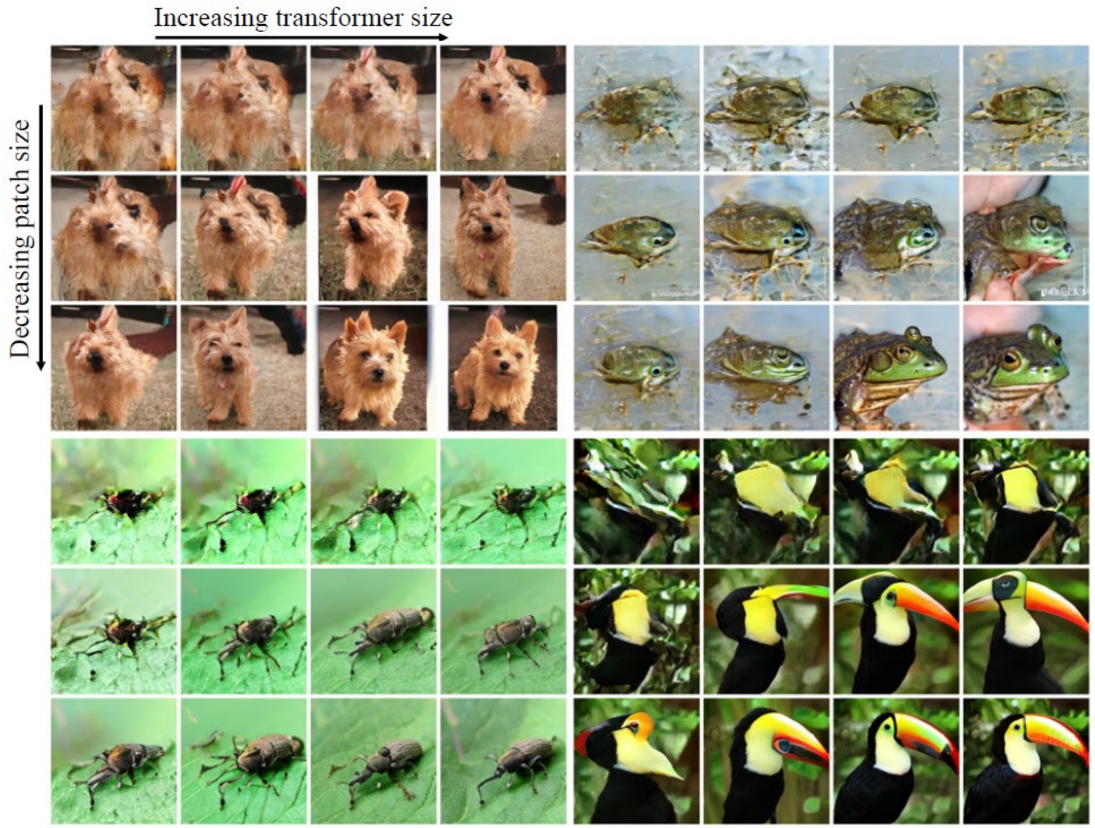


3. DiT

Results

Class-Conditional ImageNet 256×256					
Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [2]	6.95	7.36	171.4	0.87	0.28
StyleGAN-XL [53]	2.30	4.02	265.12	0.78	0.53
ADM [9]	10.94	6.02	100.98	0.69	0.63
ADM-U	7.49	5.13	127.49	0.72	0.63
ADM-G	4.59	5.25	186.70	0.82	0.52
ADM-G, ADM-U	3.94	6.14	215.84	0.83	0.53
CDM [20]	4.88	-	158.71	-	-
LDM-8 [48]	15.51	-	79.03	0.65	0.63
LDM-8-G	7.76	-	209.52	0.84	0.35
LDM-4	10.56	-	103.49	0.71	0.62
LDM-4-G (cfg=1.25)	3.95	-	178.22	0.81	0.55
LDM-4-G (cfg=1.50)	3.60	-	247.67	**0.87**	0.48
DiT-XL/2	9.62	6.85	121.50	0.67	**0.67**
DiT-XL/2-G (cfg=1.25)	3.22	5.28	201.77	0.76	0.62
DiT-XL/2-G (cfg=1.50)	**2.27**	**4.60**	**278.24**	0.83	0.57

Table 2. **Benchmarking class-conditional image generation on ImageNet 256×256.** DiT-XL/2 achieves state-of-the-art FID.



4.

DDPO

4. DDPO

Motivation

Goal: Train generative models to generate certain condition of images

satisfying Aesthetic Quality and Compressibility

Pretrained

Aesthetic Quality

Compressibility

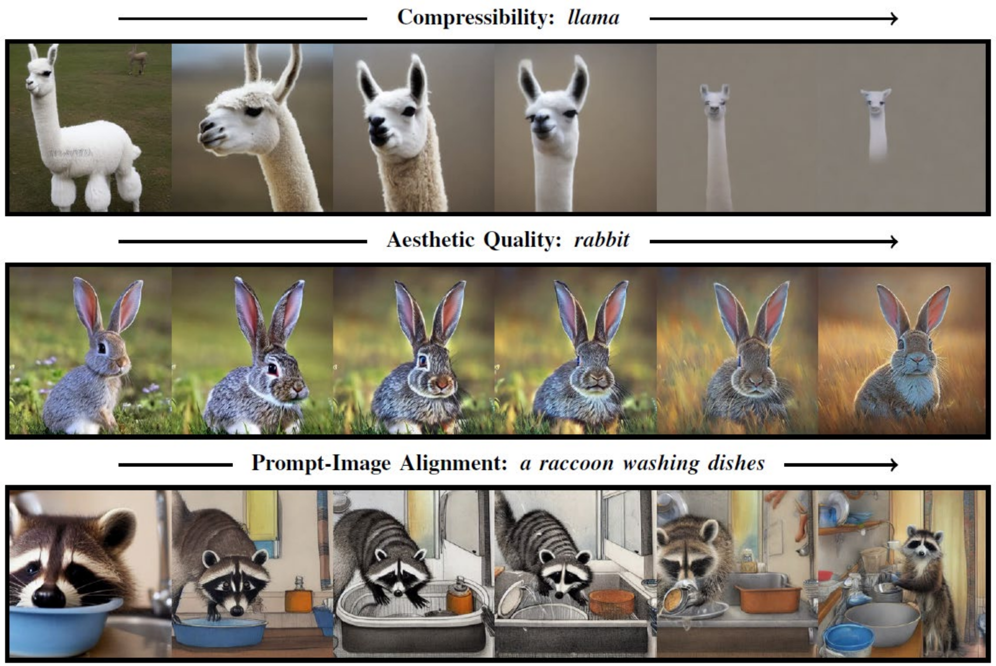
4. DDPO

Motivation

Goal: Train generative models to generate certain condition of images

satisfying Aesthetic Quality and Compressibility

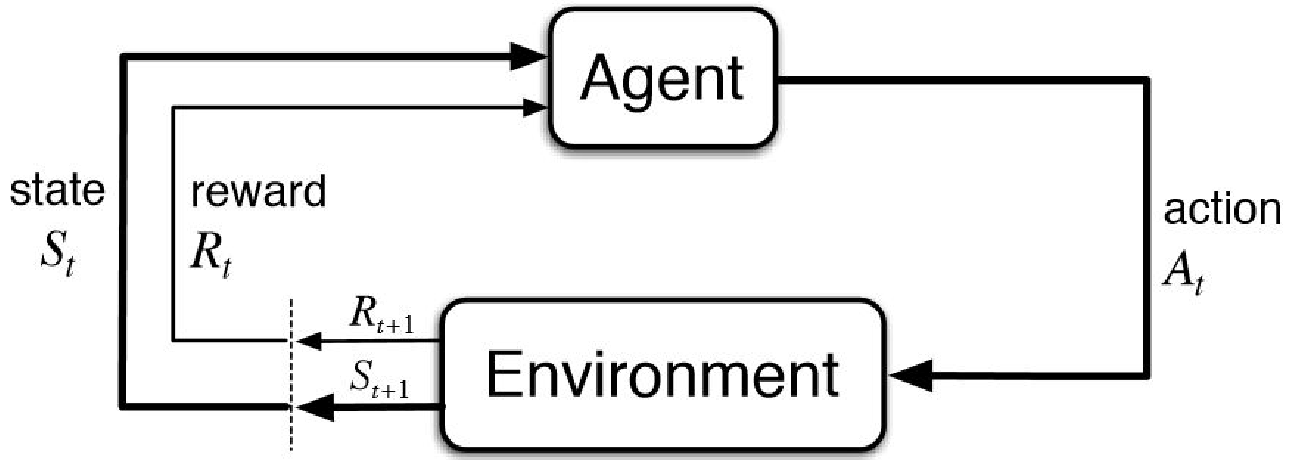
One Possible Approach: Train generative models to align given prompts using RL concept!



Motivation

Markov Decision Process (MDP)

An agent acts according to a policy $\pi(a|s)$, and trajectories are $\tau = (s_0, a_0, \ldots, s_\tau, a_\tau)$.



$$\mathcal{J}_{\text{RL}}(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T} R(\mathbf{s}_t, \mathbf{a}_t) \right]$$

4. DDPO

Diffusion as MDP

Denoising as a multi-step MDP. We map the iterative denoising procedure to the following MDP:

$$\mathbf{s}_t \triangleq (\mathbf{c}, t, \mathbf{x}_t) \quad \pi(\mathbf{a}_t \mid \mathbf{s}_t) \triangleq p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c}) \quad P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \triangleq (\delta_{\mathbf{c}}, \delta_{t-1}, \delta_{\mathbf{x}_{t-1}})$$

$$\mathbf{a}_t \triangleq \mathbf{x}_{t-1} \quad \rho_0(\mathbf{s}_0) \triangleq (p(\mathbf{c}), \delta_T, \mathcal{N}(\mathbf{0}, \mathbf{I})) \quad R(\mathbf{s}_t, \mathbf{a}_t) \triangleq \begin{cases} r(\mathbf{x}_0, \mathbf{c}) & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{J}_{\text{DDRL}}(\theta) = \mathbb{E}_{\mathbf{c} \sim p(\mathbf{c}),\ \mathbf{x}_0 \sim p_\theta(\mathbf{x}_0 \mid \mathbf{c})} \left[r(\mathbf{x}_0, \mathbf{c}) \right]$$

1) **State**: Condition (Context, text) $\boldsymbol{c}$ + Diffusion time step t + Image at time step t $\boldsymbol{x}_t$

2) **Action**: Denoised Image $\boldsymbol{x}_{t-1}$

3) **Reward**: Only computed in the final image by the given condition $r(\boldsymbol{x}_0, \boldsymbol{c})$

→ **Optimized by policy gradient** estimation

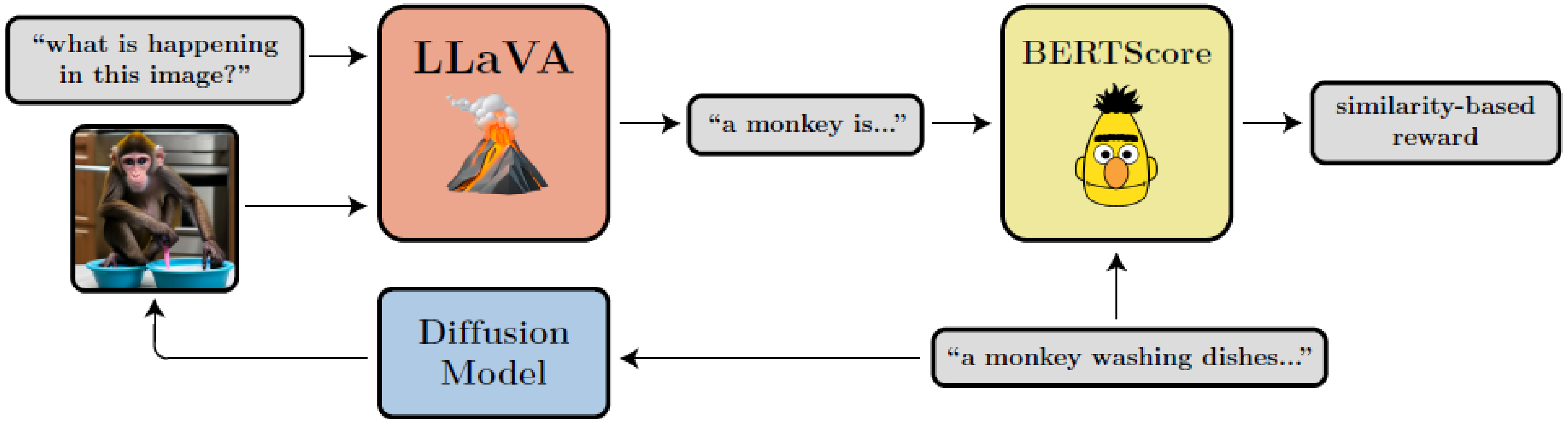
Diffusion as MDP

Ways to set the reward

Aesthetic Quality: LAION aesthetics predictors (trained on 176,000 human ratings)

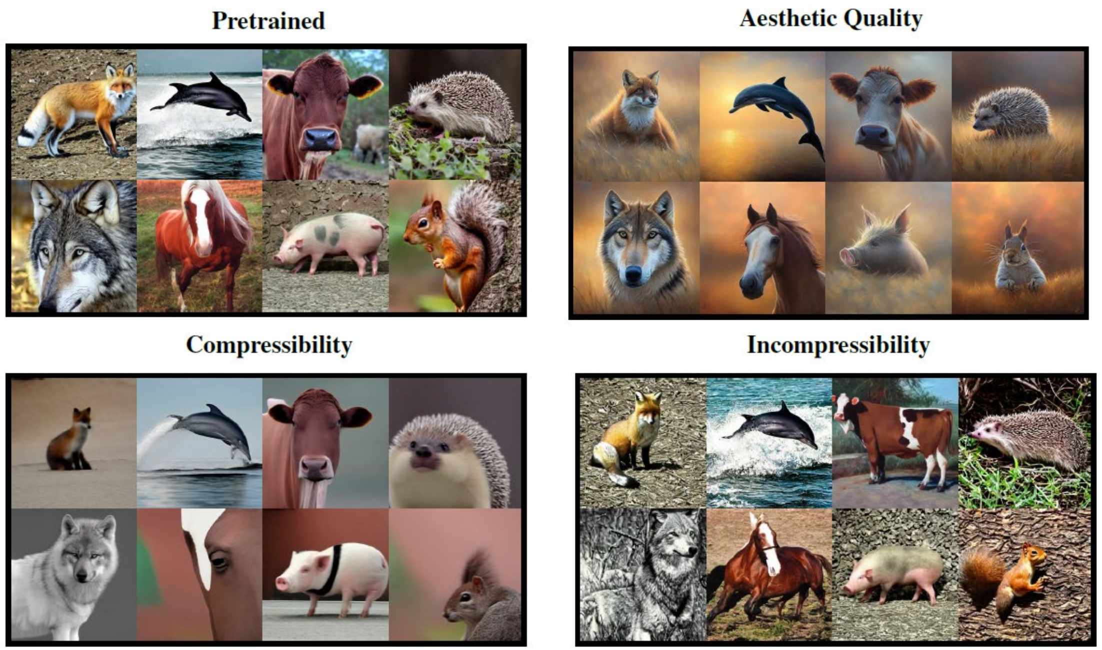
Compressibility: File size of the image after JPEG compression

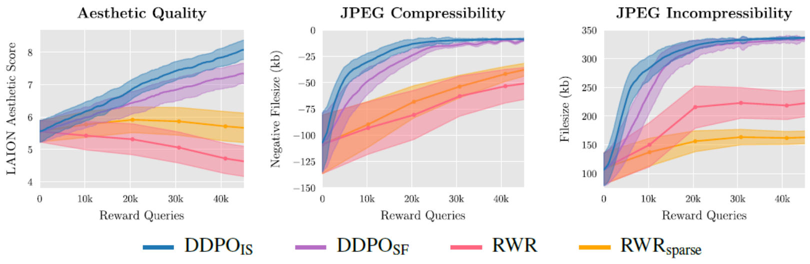
Prompt Alignment: Prompt comparison with Vision-Language Model's generated image caption

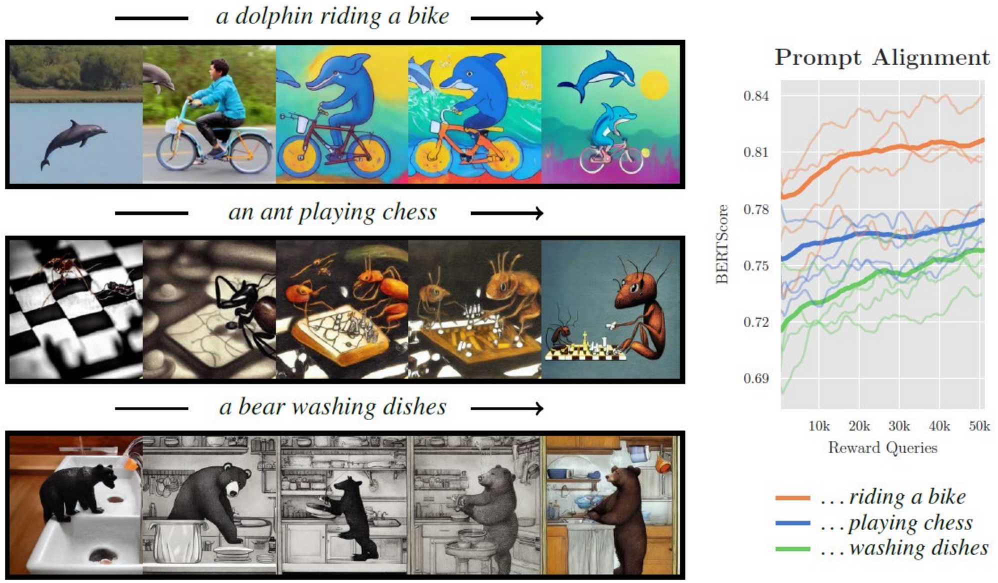


4. DDPO

Results







4. DDPO

Results

5.

Summary

5. Summary

Summary

Diffusion Model

DDPM (2020)

Denoising noise assuming Gaussian

LDM (2021)

Diffusion + VAE (encoder-decoder)

DiT (2022)

Diffusion + Vision Transformer

DDPO (2023)

Diffusion + Reinforcement Learning

So, what can we do next for Diffusion?

5. Summary

Reference

Ho et al., Denoising Diffusion Probabilistic Models, 2020. (NeurIPS 2020)

Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models, 2021. (CVPR 2022)

Peebles et al., Scalable Diffusion Models with Transformers, 2022. (ICCV 2023)

Black et al., Training Diffusion Models with Reinforcement Learning, 2023. (ICLR 2024)

Kyungwoo Song, STA3145 <Reinforcement Learning> Lecture Note, Spring 2024.

YONSEI
ARTIFICIAL INTELLIGENCE

24-Fall 생성논문팀 분리세션

발제자: 14기 김민규 (kimmin01@yonsei.ac.kr)