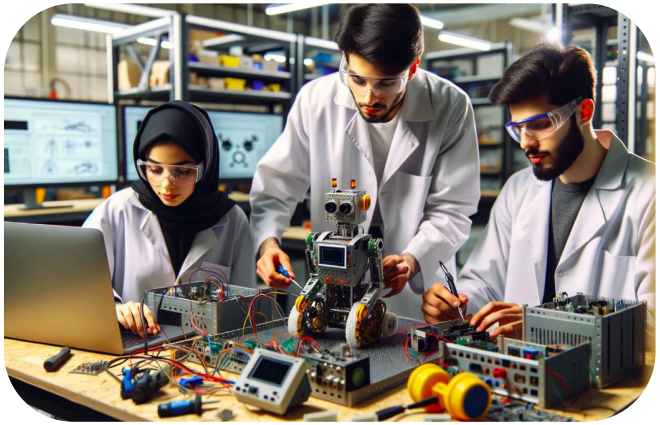
Some Applications of Diffusion Models



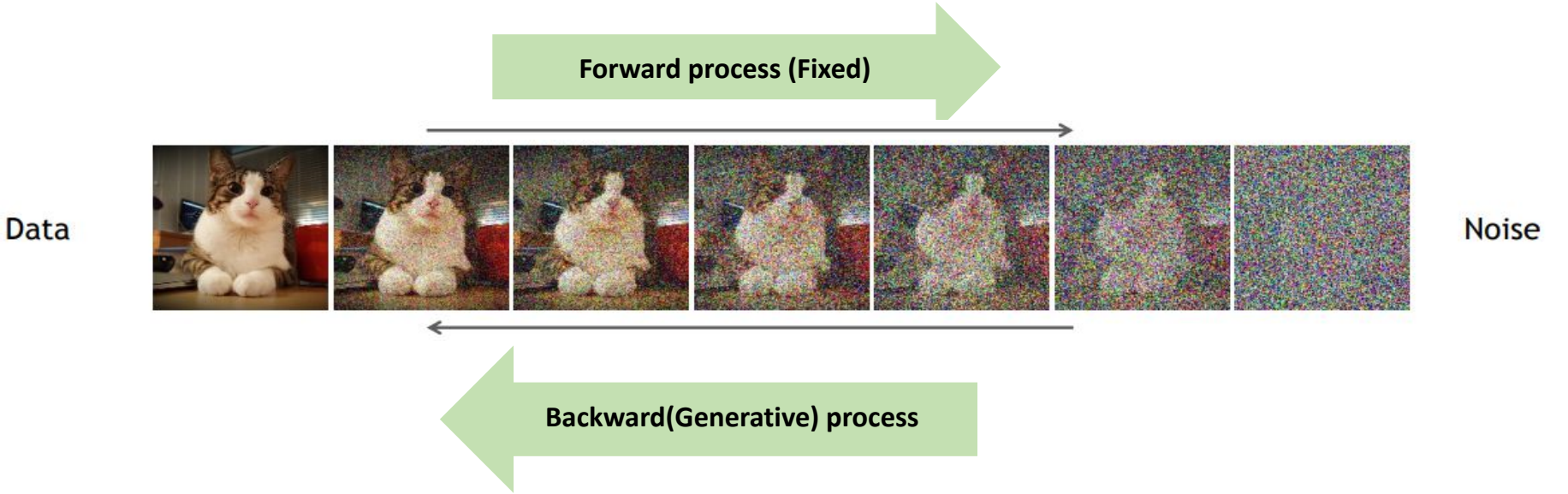
Farshad Sangari
farshads7778@gmail.com

Table of content

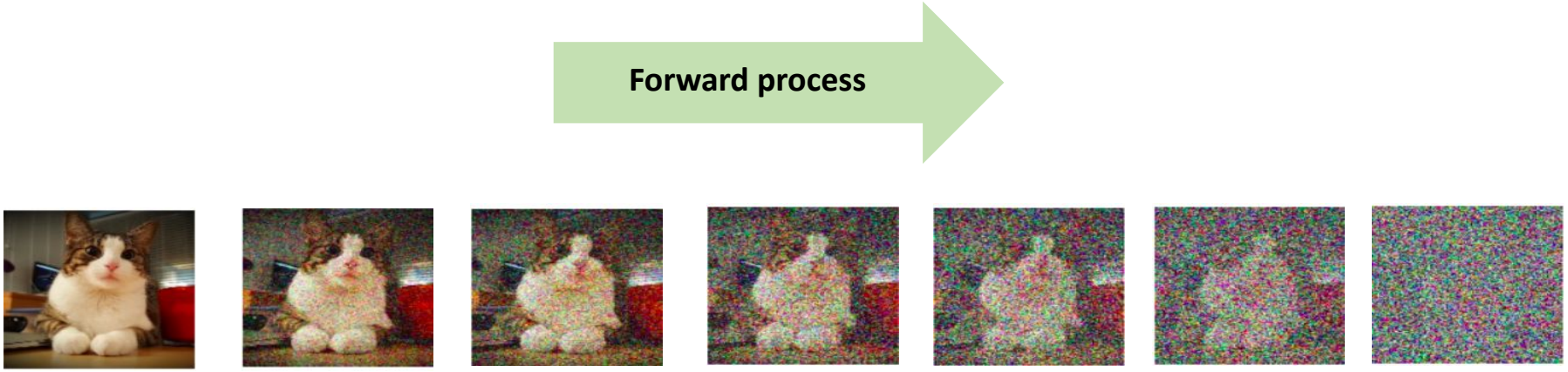
- Deep Diffusion Probabilistic model (DDPM)
- Latent Diffusion model (LDM)
- Diffusion Autoencoder (DiffAE)
- Classifier-guided Sampling Method
- Unsupervised Representation Learning from Pre-trained DPM(PDAE)

Deep Diffusion Probabilistic model (DDPM)

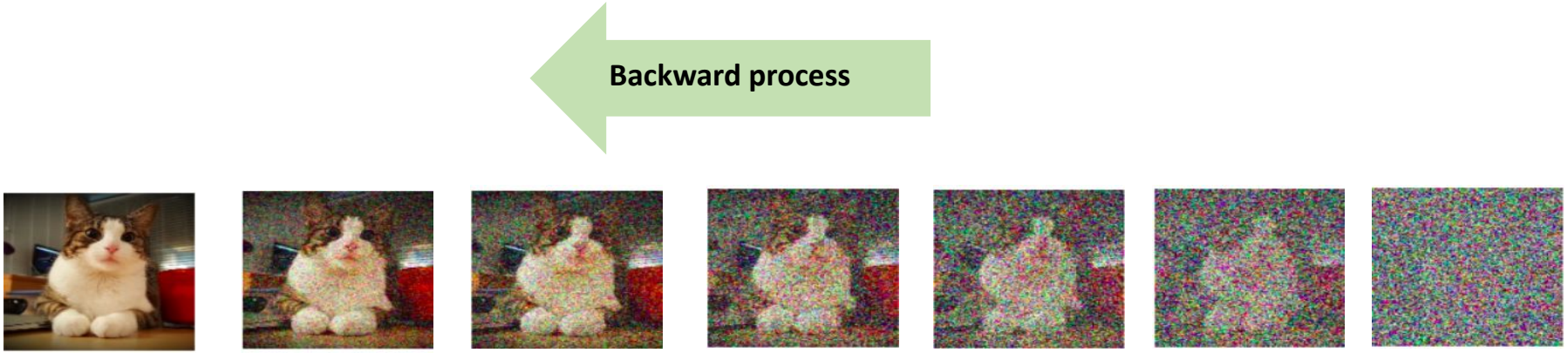
Overview

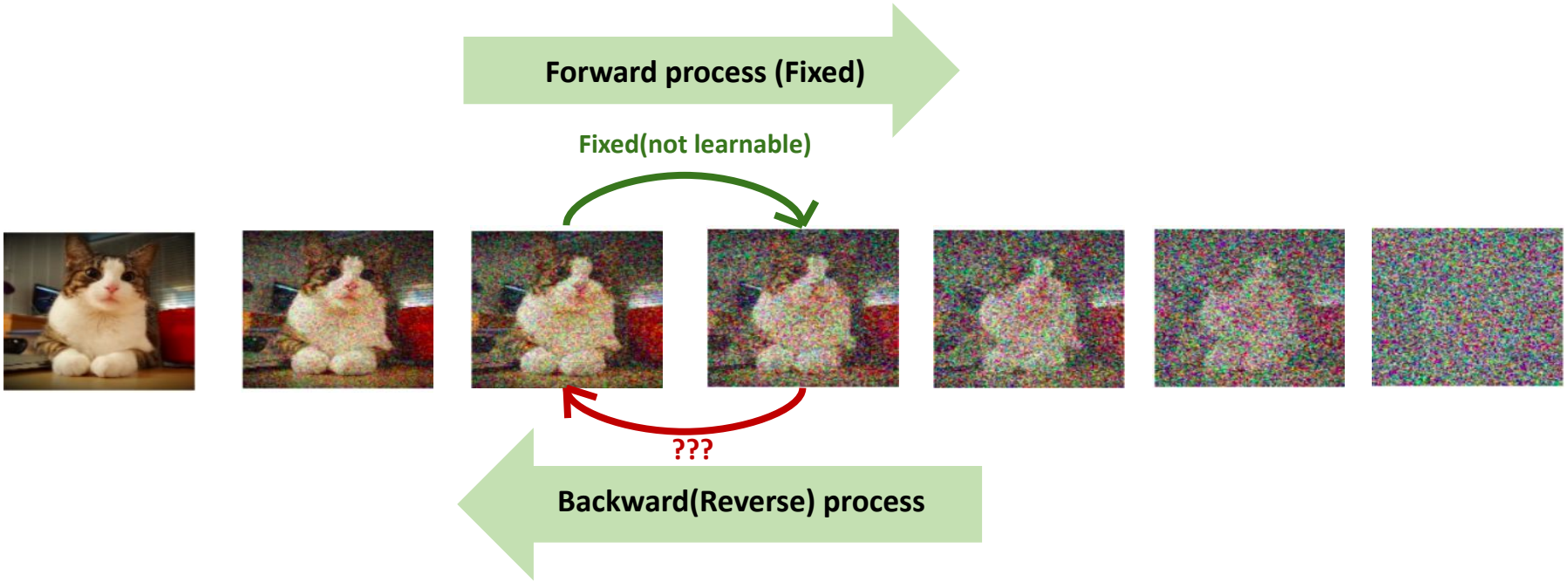

Forward process (Fixed)
Data
Noise
Backward(Generative) process

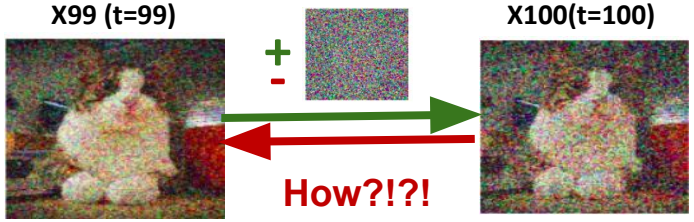
Forward process (Fixed)



Reverse process(Generative)



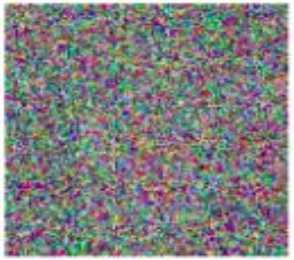

Forward process (Fixed)
Fixed(not learnable)
???
Backward(Reverse) process



X100

t=100

ε100



X99 (t=99)

Time Representation

$$L_{simple} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0),\, \epsilon \sim \mathcal{N}(0,\mathrm{I}),\, t \sim U(1,T)} \left[\| \epsilon - \epsilon_\theta \left(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon}_{\mathbf{x}_t}, t \right) \|^2 \right]$$

Generation

Challenges?!

Challenges

Speed of sampling

- Deep Diffusion Implicit Model(DDIM)
- Consistency Models

Need strong resources

✅ • Latent Diffusion model (LDM)

Having more control on generation process(no meaningful latent available!)

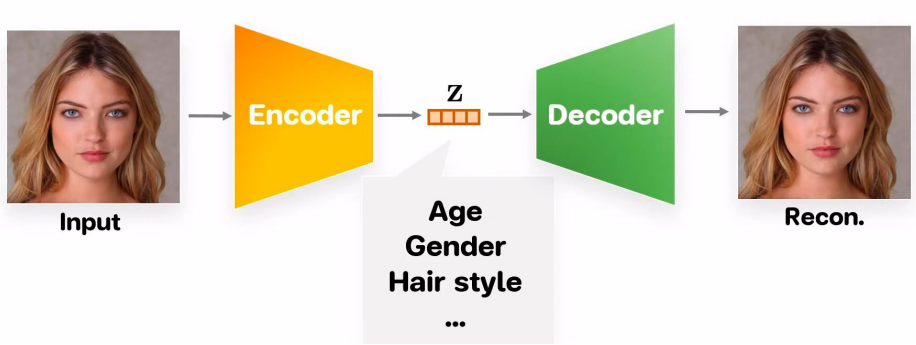
✅ • Diffusion Autoencoder (DiffAE)

✅ • Classifier-guided Sampling Method

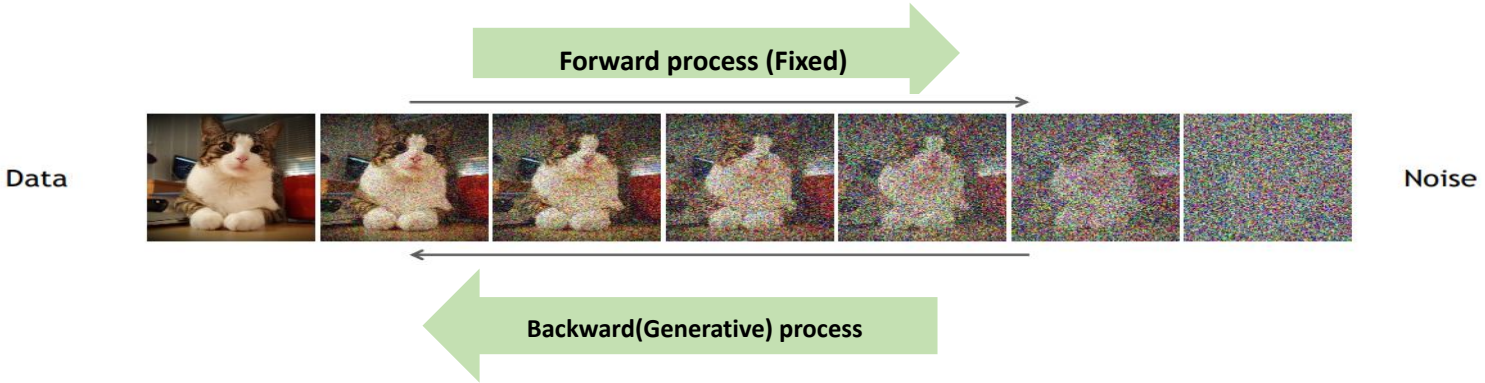
✅ • Unsupervised Representation Learning from Pre-trained DPM(PDAE)

DDPM vs VAE

VAE:



DDPM:



Challenges

Speed of sampling

- Deep Diffusion Implicit Model(DDIM)
- Consistency Models

Need strong resources

✅ • Latent Diffusion model (LDM)

Having more control on generation process(no meaningful latent available!)

✅ • Diffusion Autoencoder (DiffAE)

✅ • Classifier-guided Sampling Method

✅ • Unsupervised Representation Learning from Pre-trained DPM(PDAE)

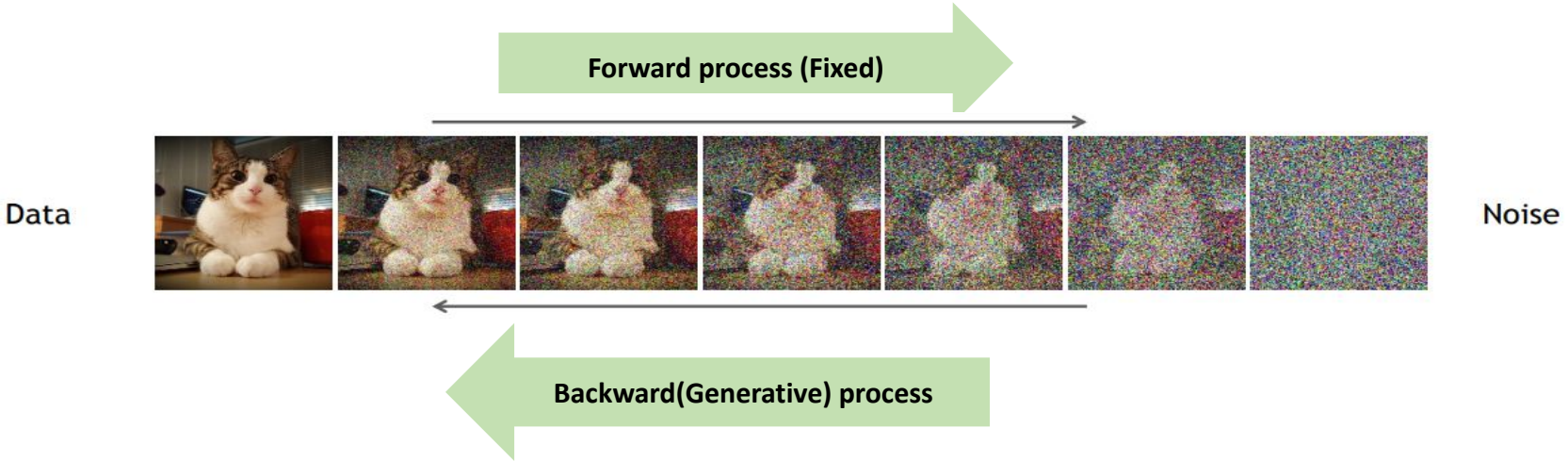
Latent Diffusion Model

Contributions?!

- How can we reduce the required resources?
- How to obtain conditional Model?

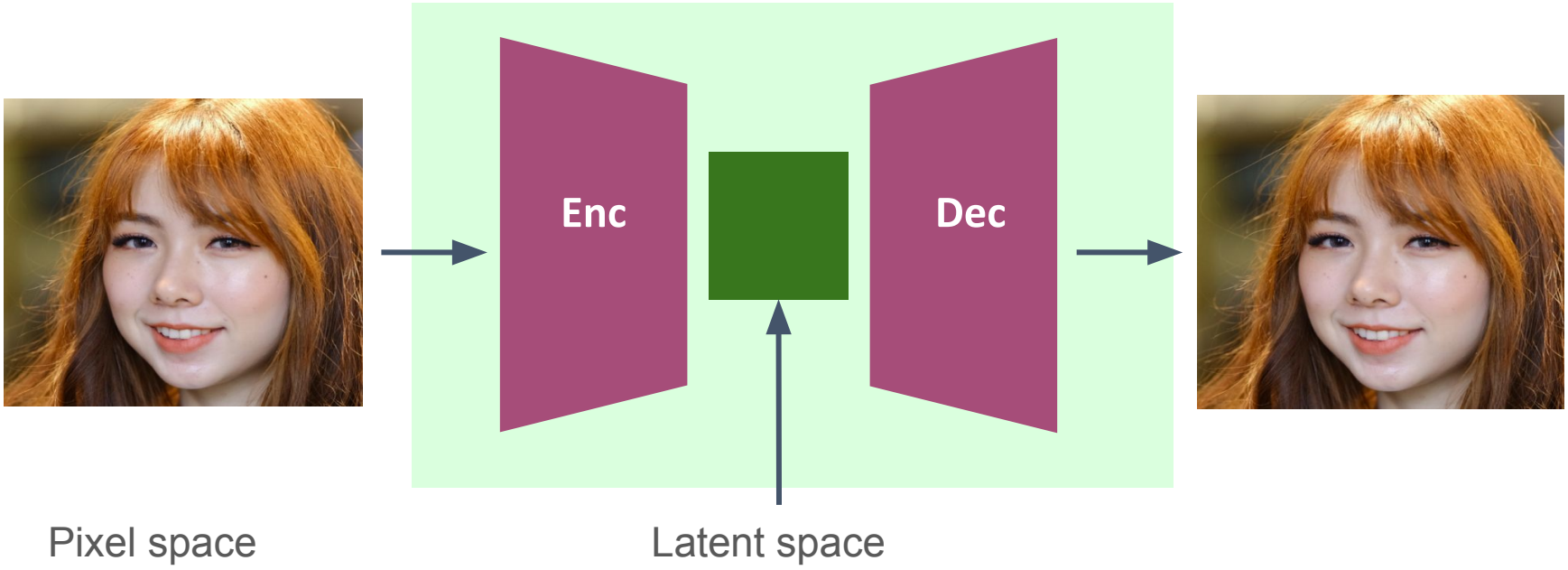
Review DDPM

- In conventional diffusion models, datas are in pixel space(original space)



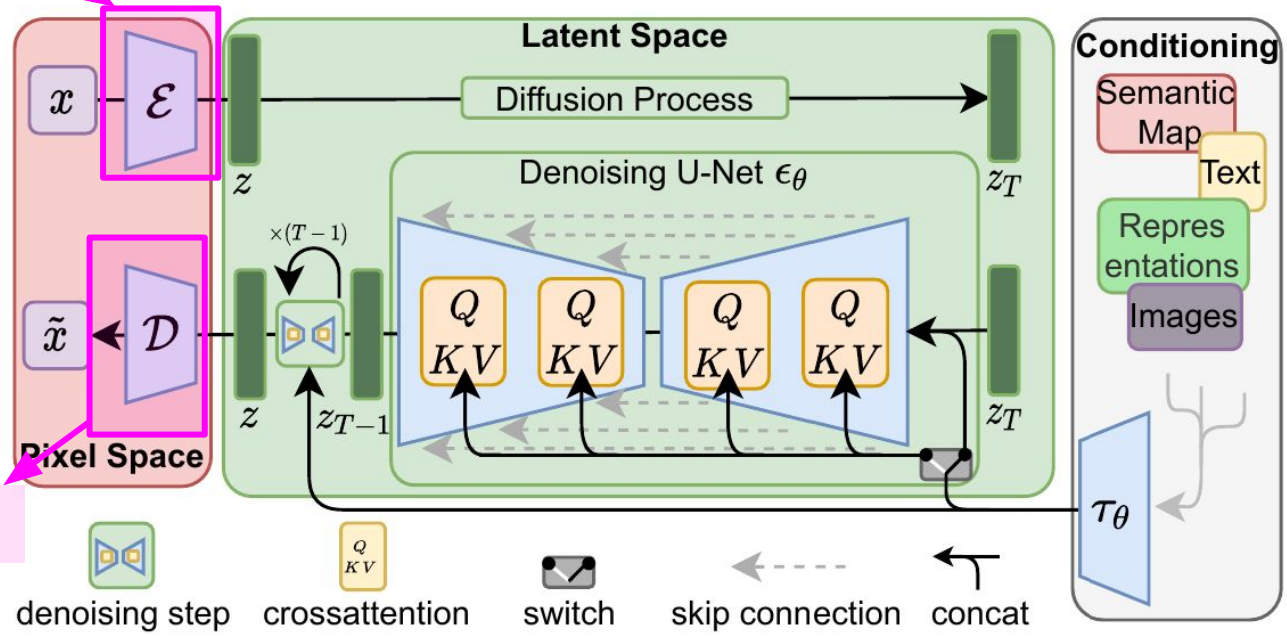
Idea

- How about to use feature map instead of pixel space representations?!



Overview

Brings us to latent space



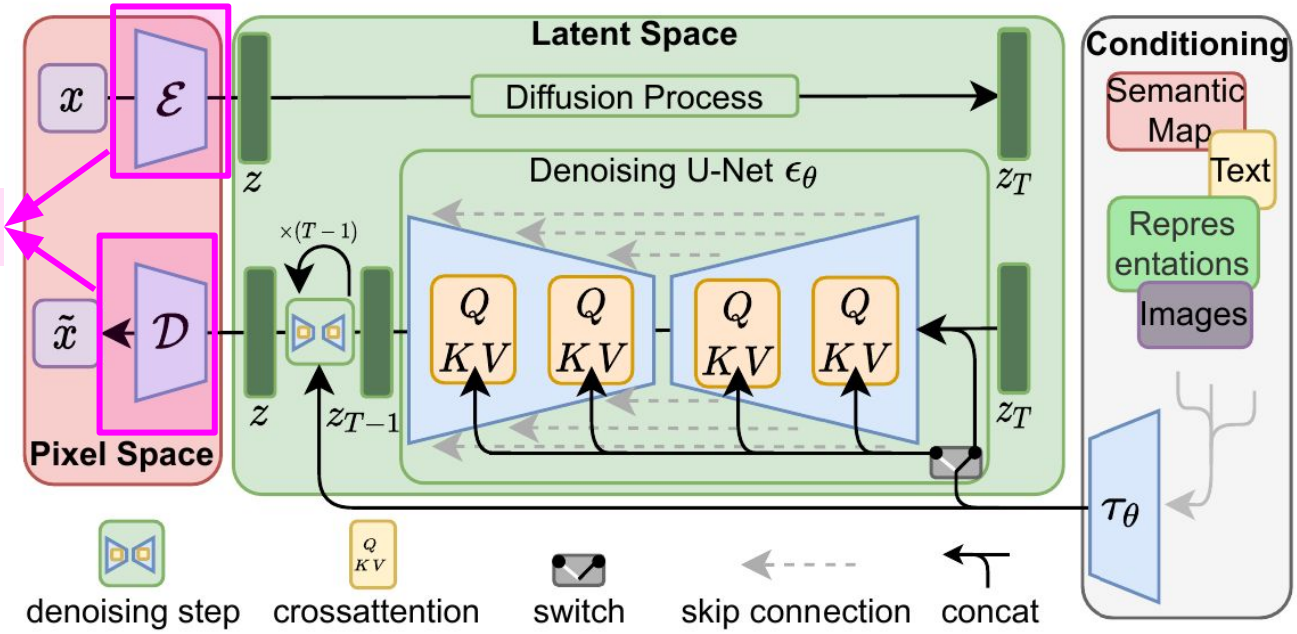
Drops us to Pixel space

Training steps

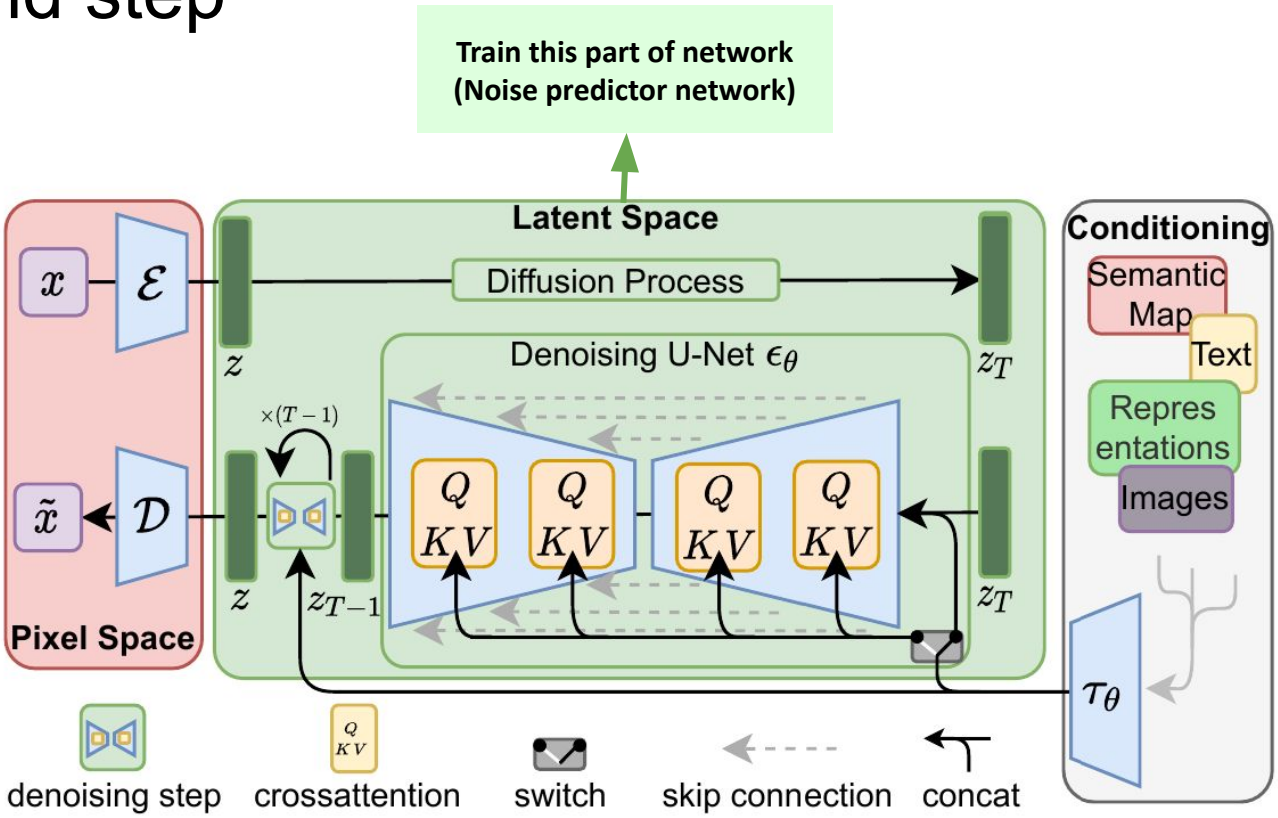
- First train such encoder and decoder
- Second train regular diffusion(DDPM) in latent space

First step

Train this part of network



Second step


Train this part of network
(Noise predictor network)
Latent Space
Diffusion Process
Denoising U-Net ϵ_θ
$\times(T-1)$
x
$\mathcal{E}$
$\tilde{x}$
$\mathcal{D}$
Pixel Space
z
z_T
z_{T-1}
Q
KV
τ_θ
Conditioning
Semantic Map
Text
Repres entations
Images
denoising step
crossattention
switch
skip connection
concat

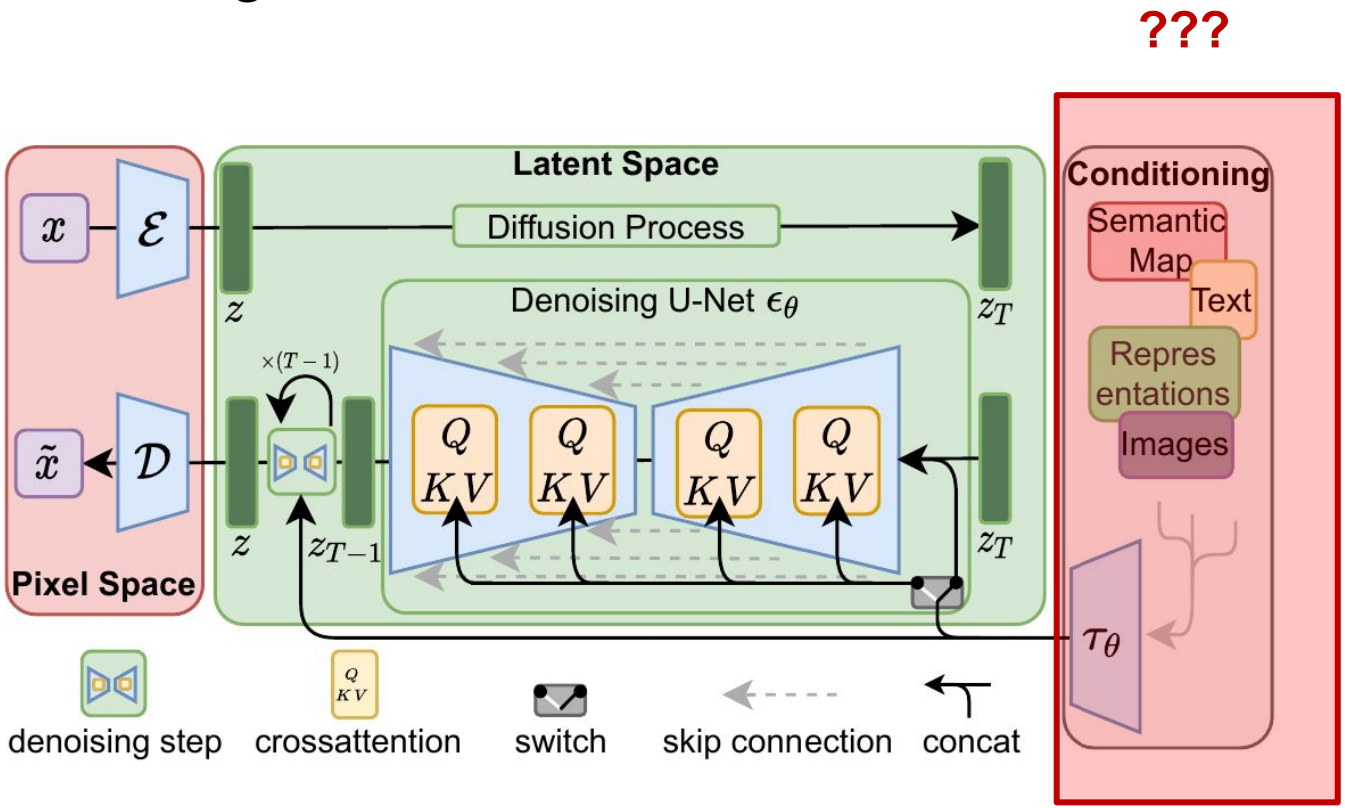
Second step

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

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

z

Conditioning



Conditional applications

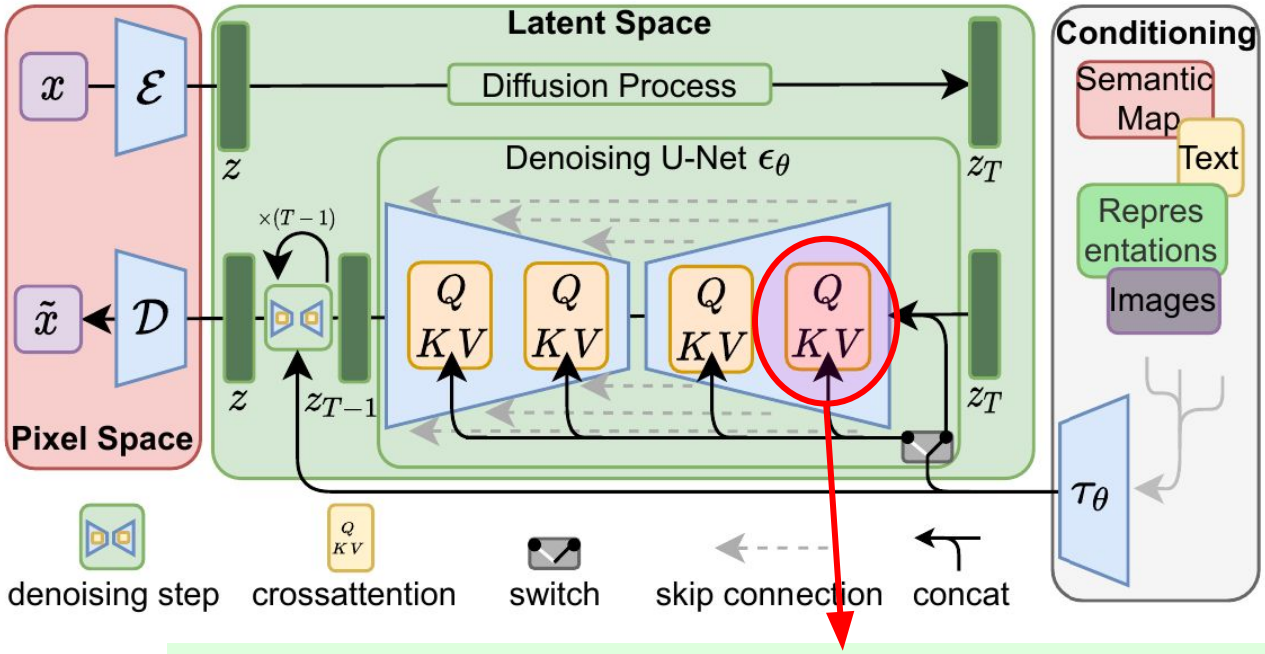
DALL·E 2

"a teddy bear on a skateboard in times square"

Imagen

A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

Conditioning



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

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

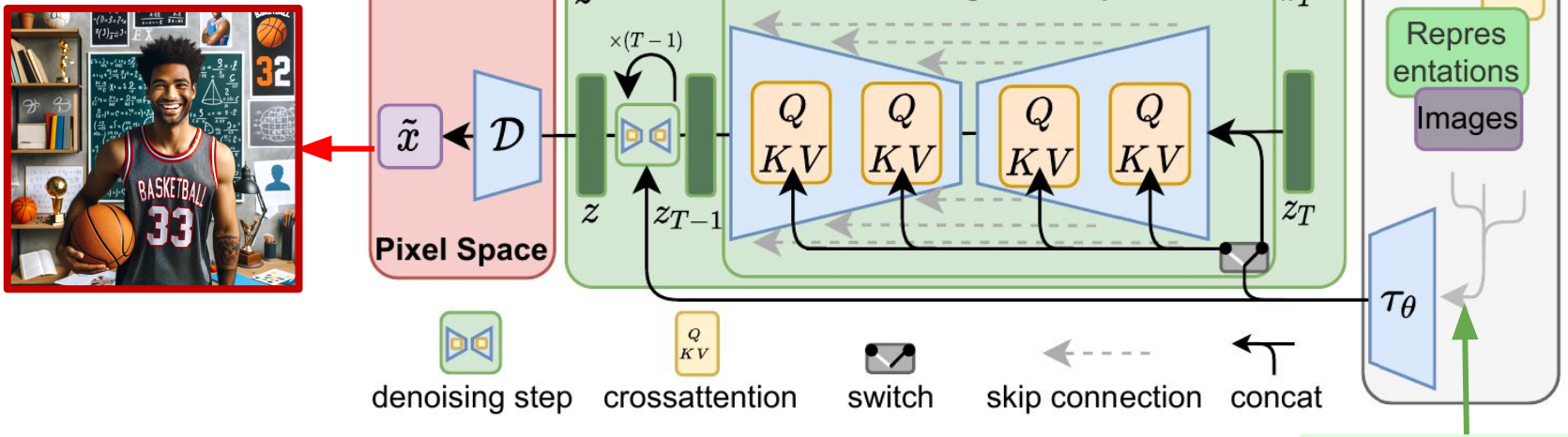
Loss Function (Conditional mode)

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

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

Some Samples by DALL·E 2

How can we sample from such network?!

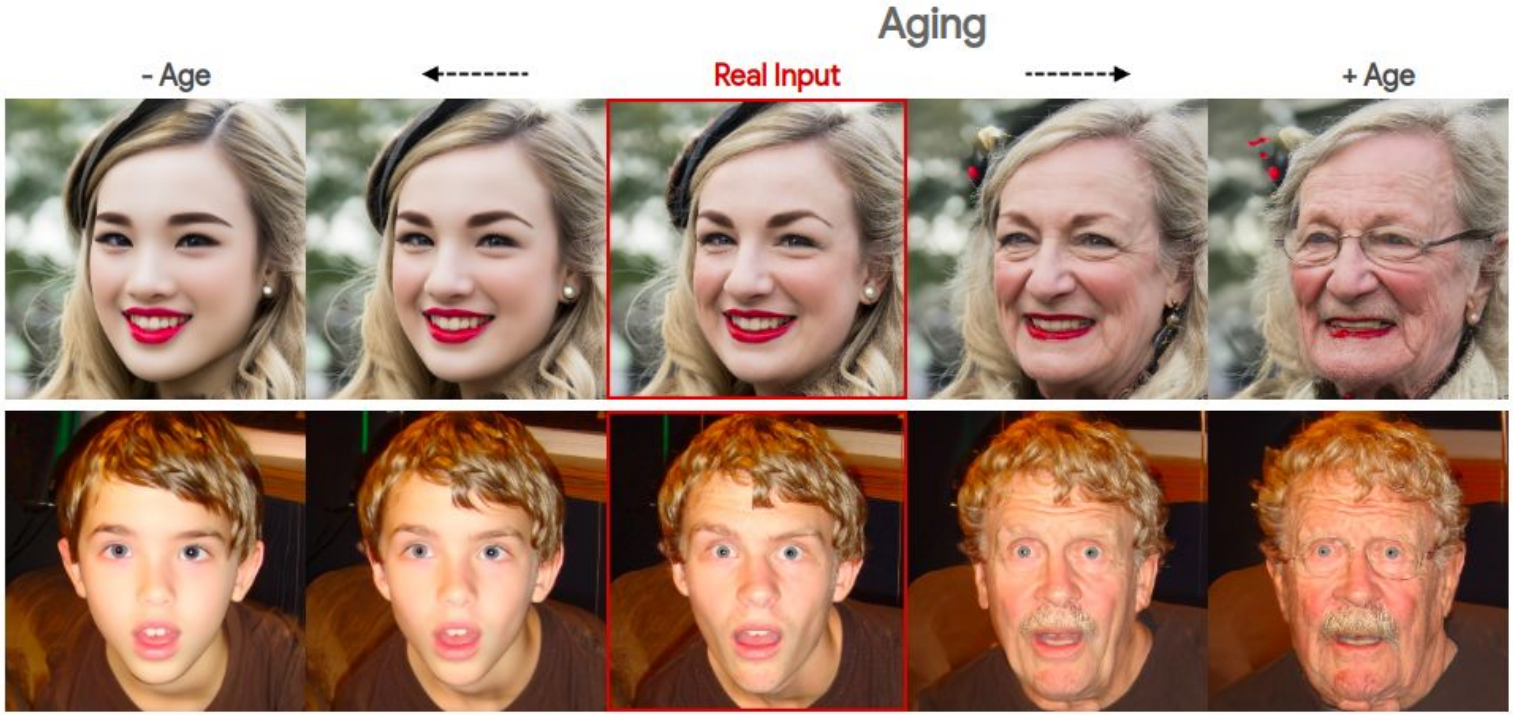


Diffusion Autoencoder

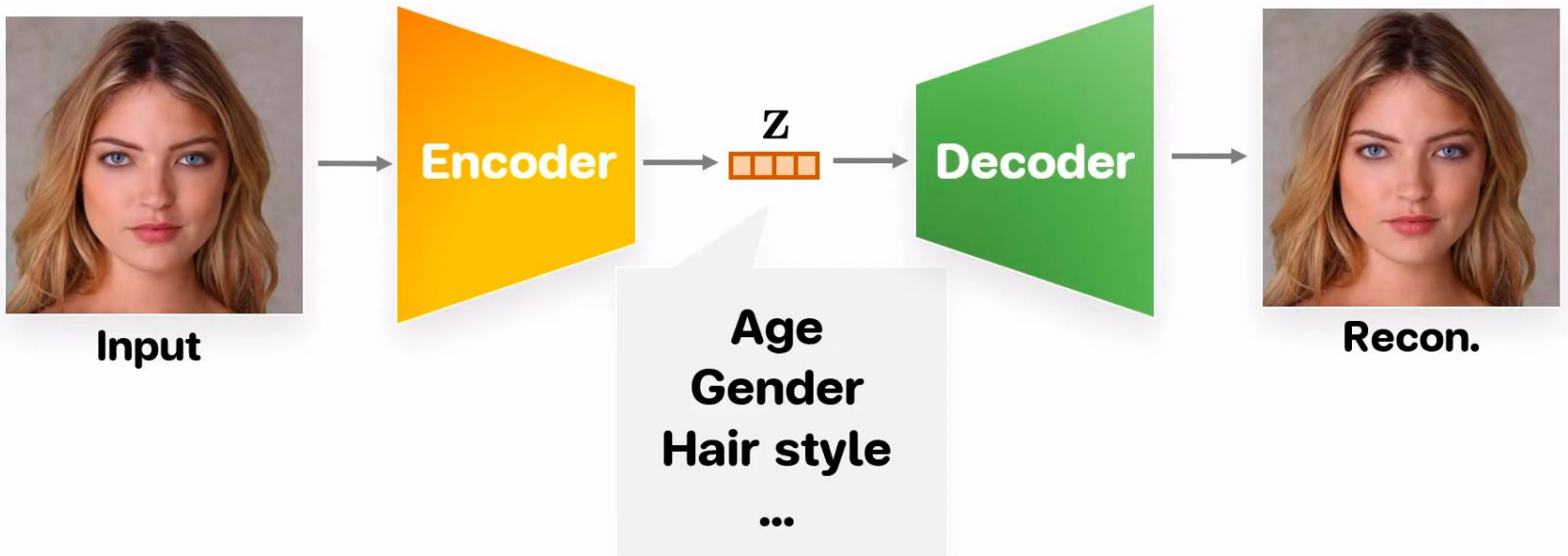
Contributions?!

- How to have a meaningful latent space?

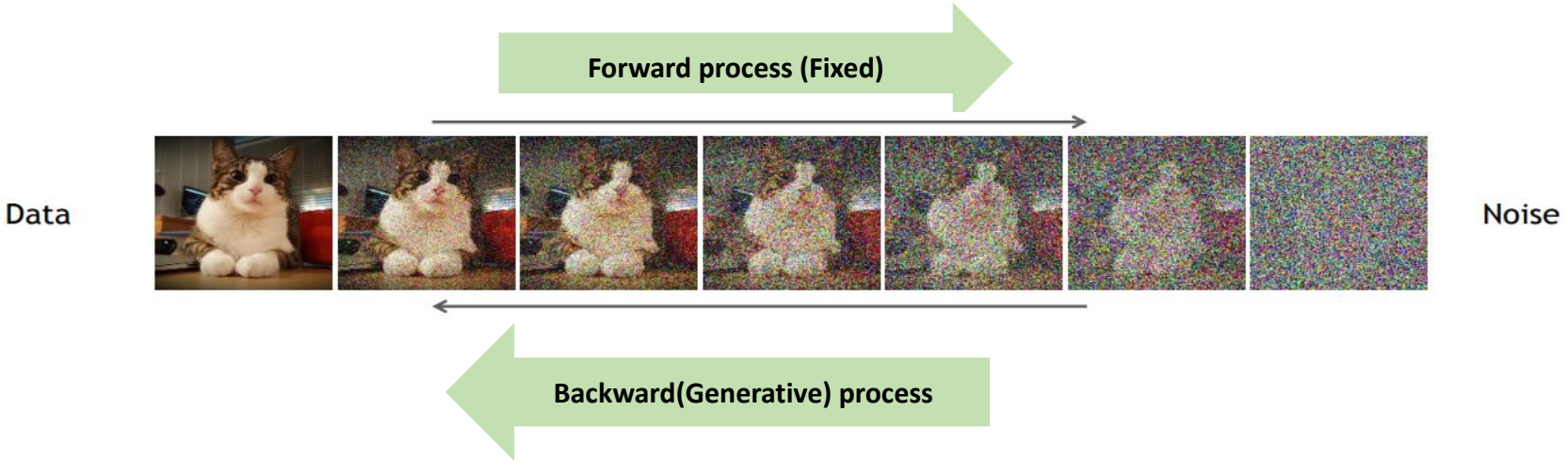
Problem?



How was VAE?

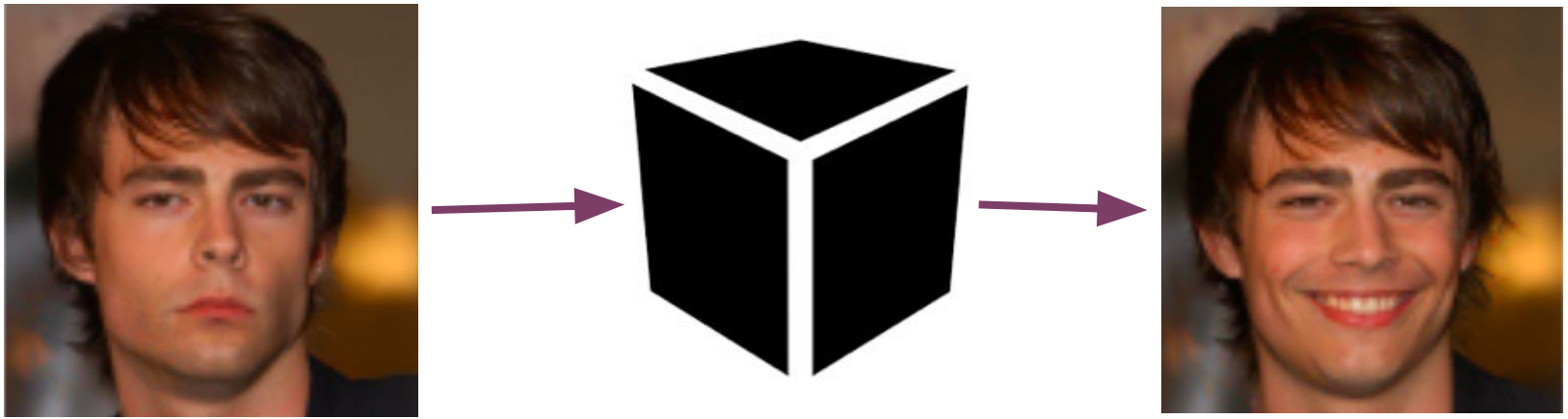


Review(DDPM)


Forward process (Fixed)
Data
Noise
Backward(Generative) process

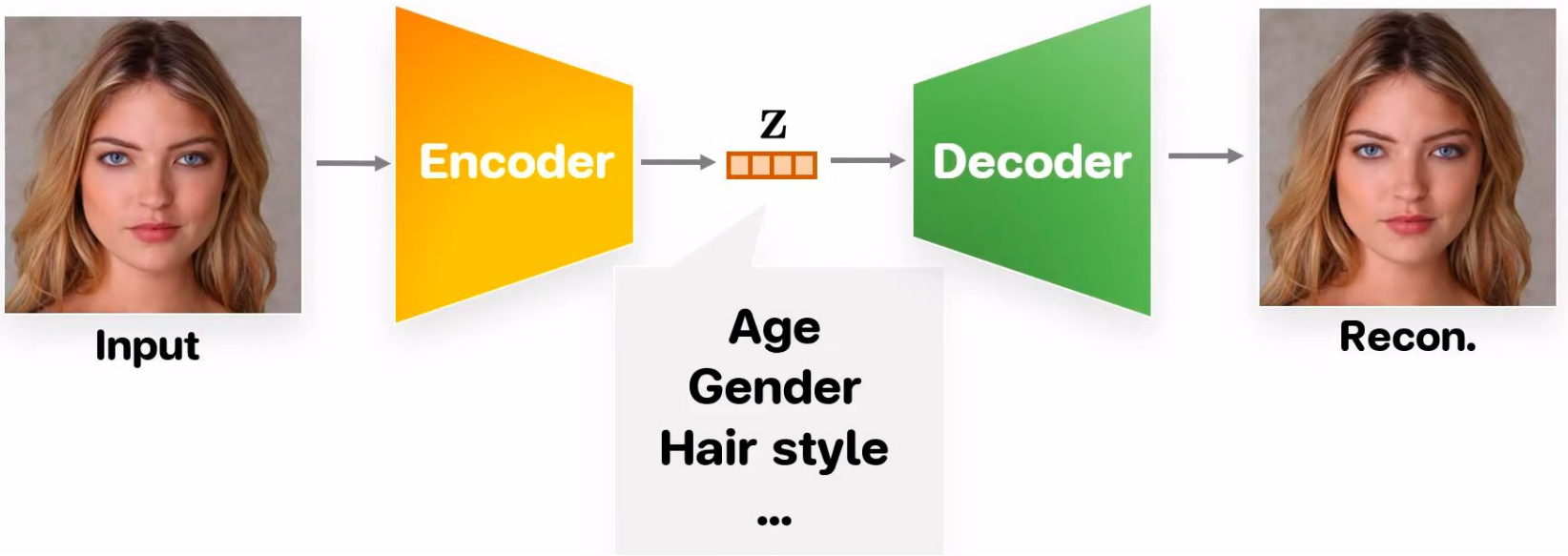
What is the problem with DDPM?

Diffusion model



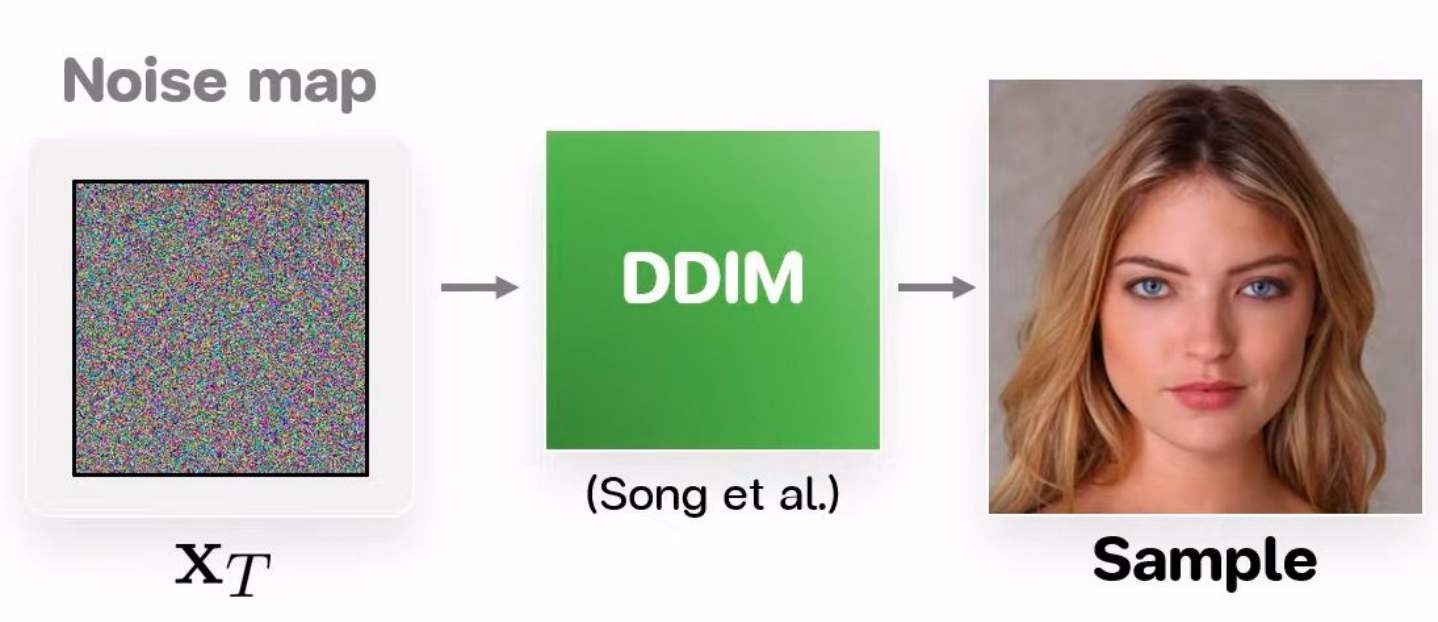
How to obtain such model?!

Step zero: How was VAE?



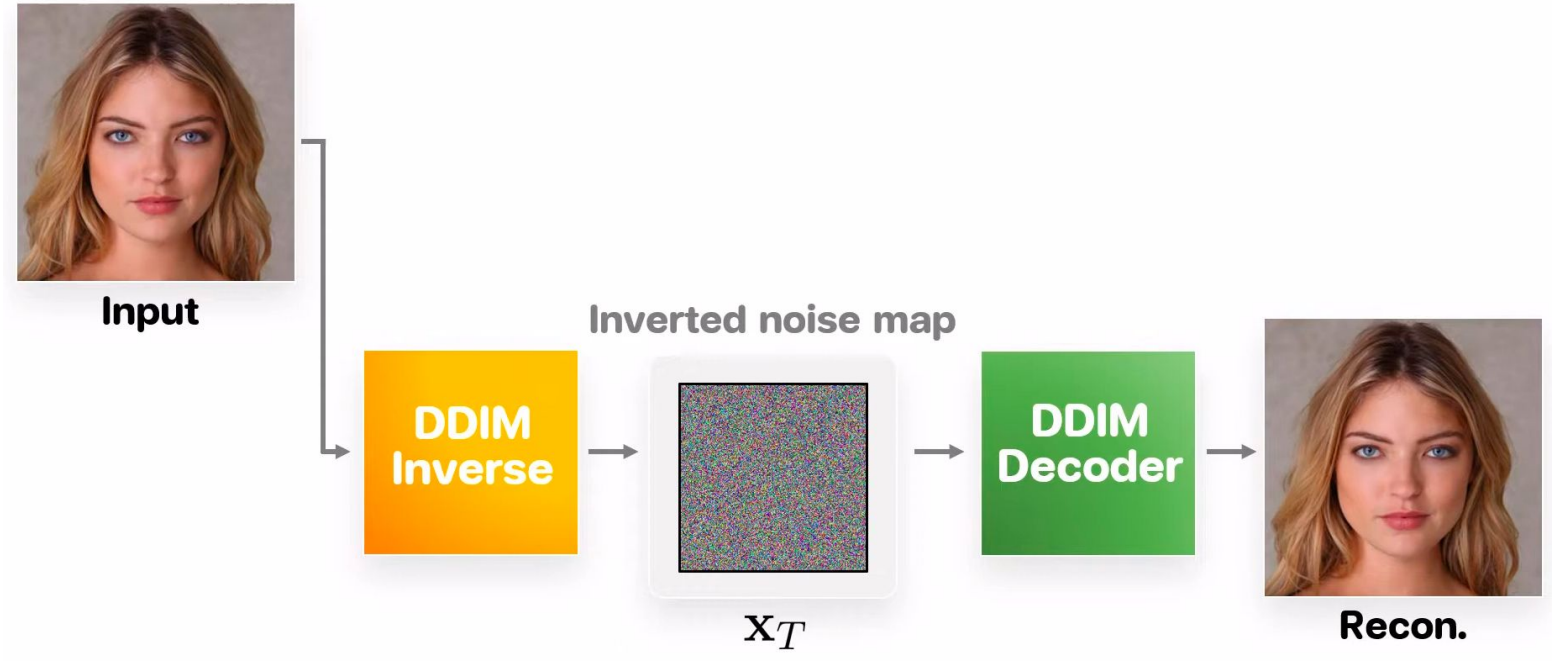
How to obtain such model?!

Step one: Convert DDIM to VAE (somehow)



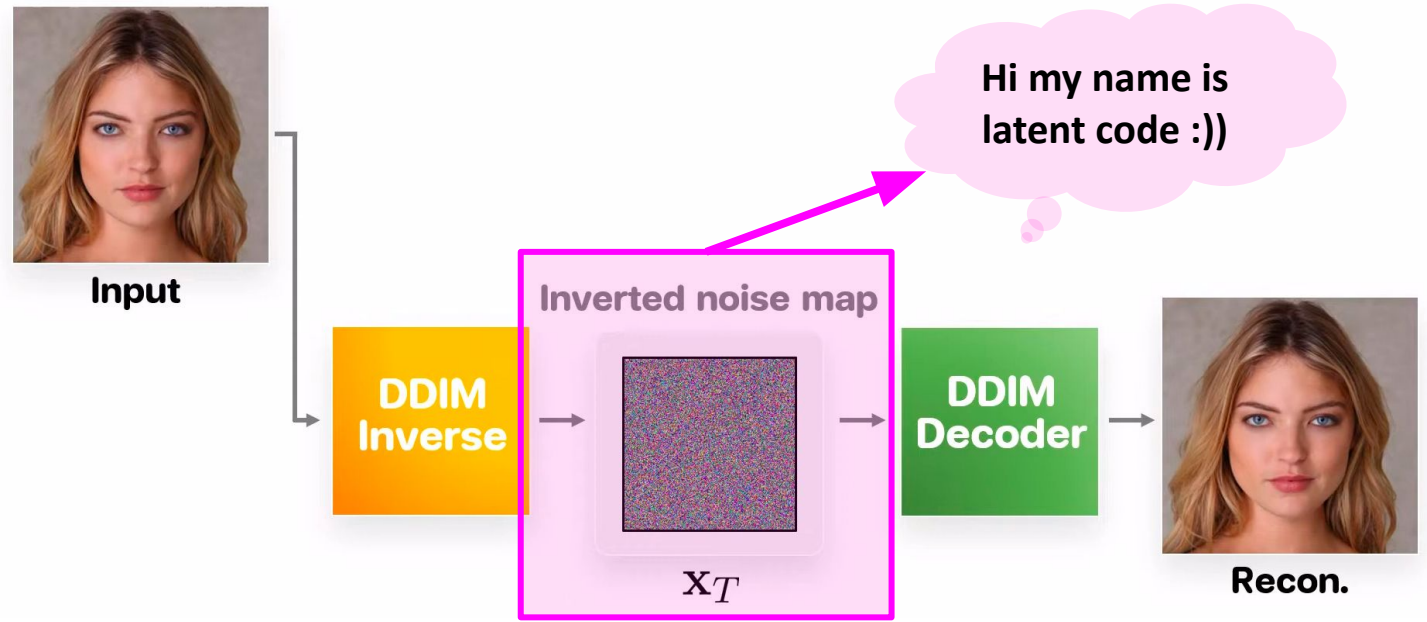
How to obtain such model?!

Step one: Convert DDIM to VAE (somehow)



How to obtain such model?!

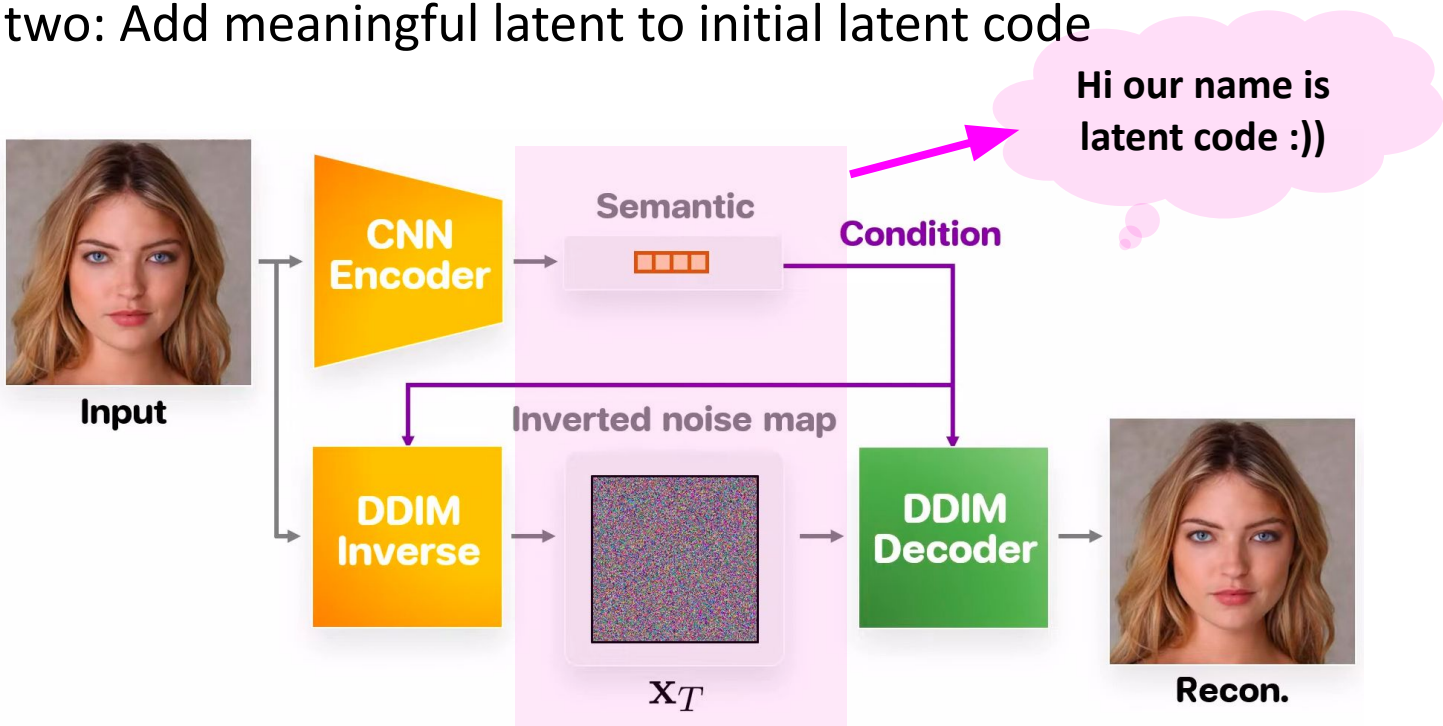
Step one: Convert DDIM to VAE (somehow)



Now we reach this purpose, but the latent code have no meaning!!

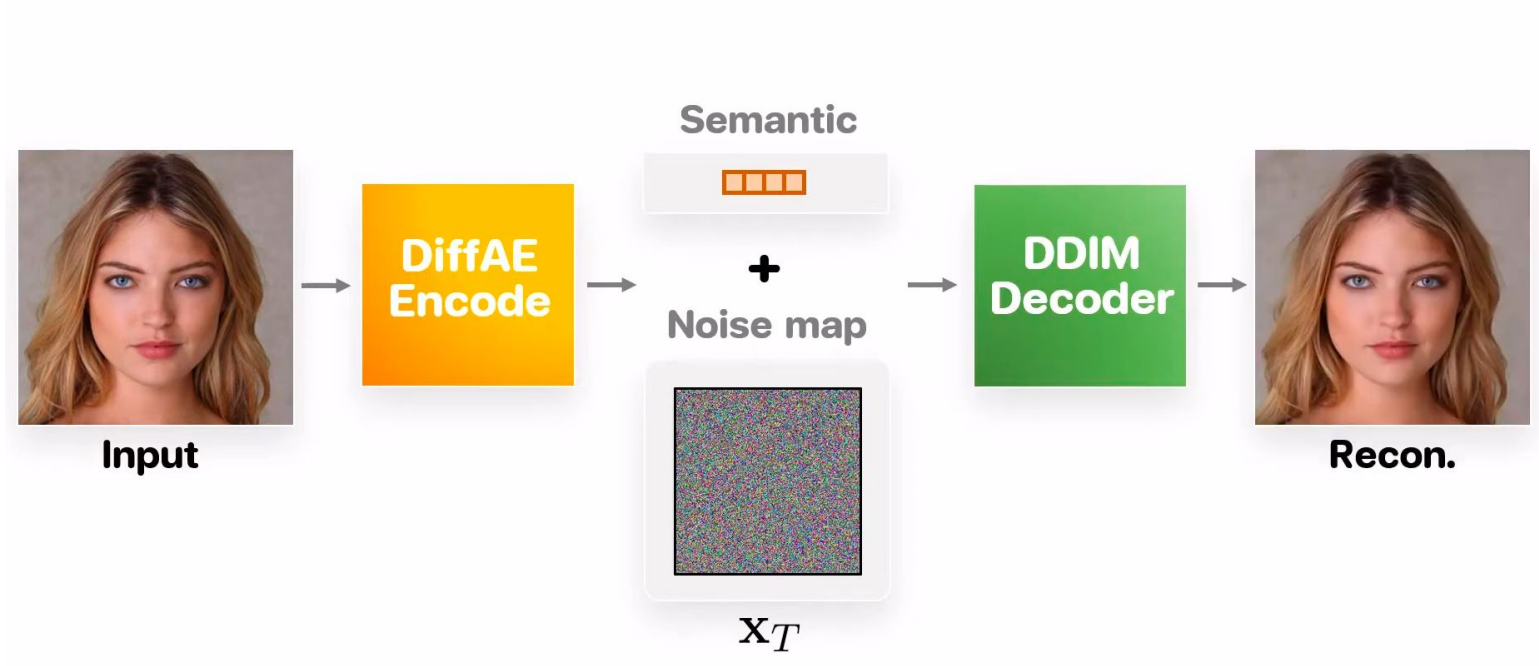
How to obtain such model?!

Step two: Add meaningful latent to initial latent code

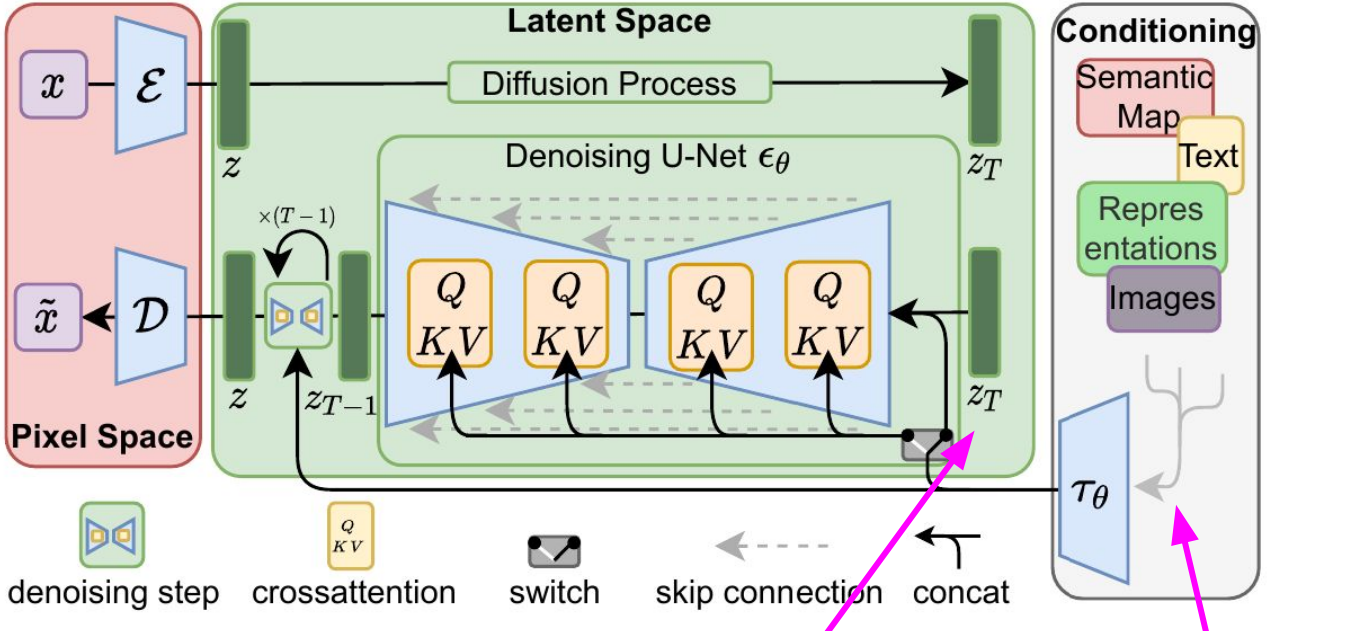


How to obtain such model?!

Step two: Add meaningful latent to initial latent code

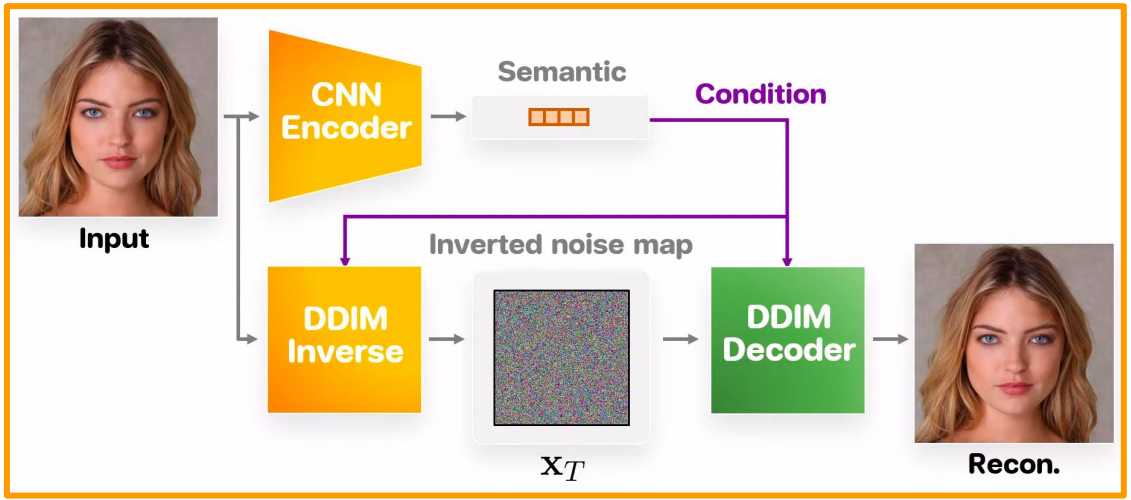


How does decoder works?



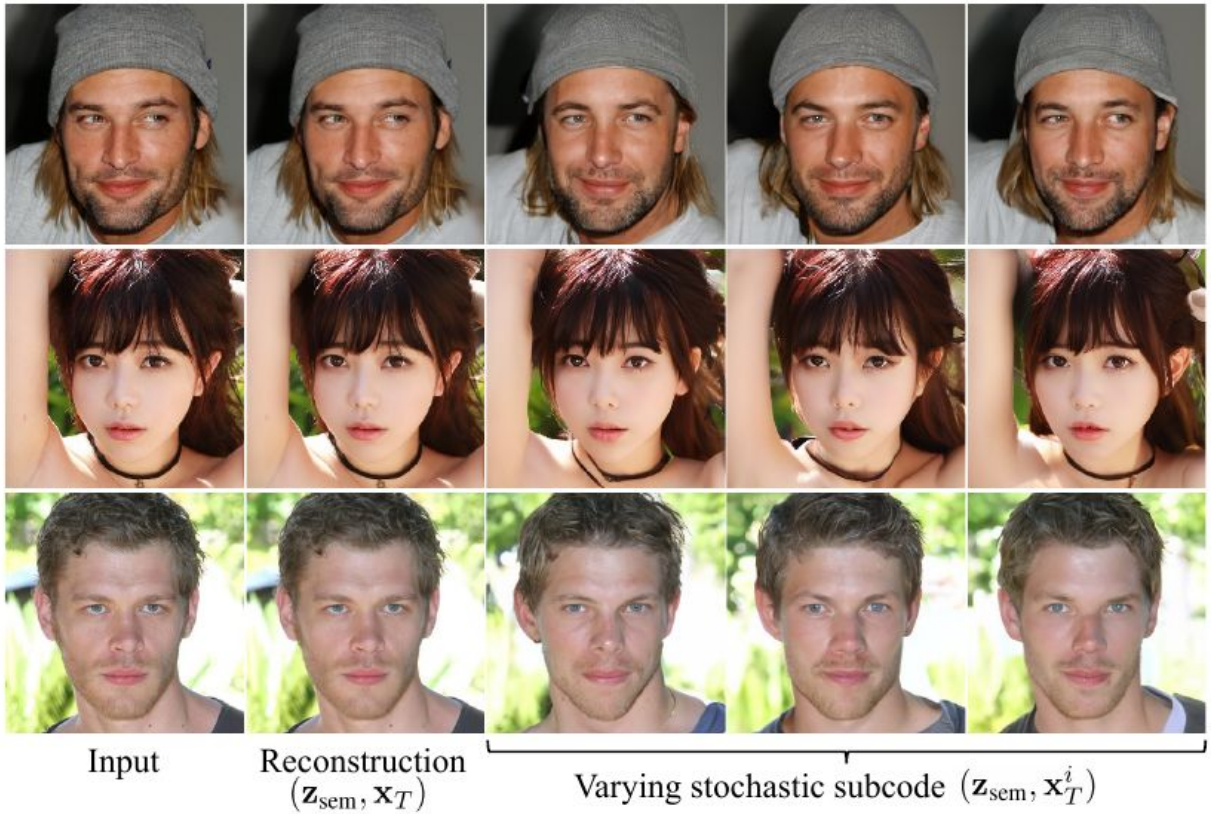
Training For Conditional mode

- Train the semantic encoder (φ) and the image decoder (θ) until convergence



$$L_{\text{simple}} = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0,\epsilon_t}\left[\|\epsilon_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}}) - \epsilon_t\|_2^2\right]$$

Experiments (Changing stochastic code(x_T))



Experiments (Changing semantic code(z_{sem}))

Classifier-guided Sampling Method

Contributions?

- Conditional sampling with a **pretrained** DDPM

Classifier-guided **Sampling** Method

Goal: Guide some pre-trained unconditional DDPM to sample towards specified class y

This paper tries to achieve this goal without training DPM

Review DDPM

$$L_{simple} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0),\, \epsilon \sim \mathcal{N}(0,\mathrm{I}),\, t \sim U(1,T)} \left[\| \epsilon - \epsilon_\theta \left(\underbrace{\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon}_{\mathbf{x}_t}, t \right) \|^2 \right]$$

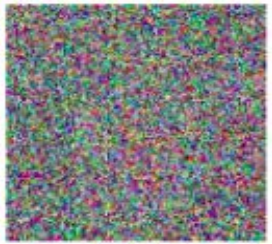
X100

t=100

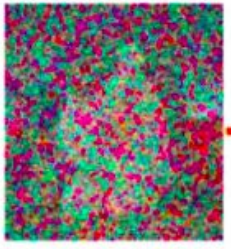
Time Representation

A network like UNET

ε100



X99 (t=99)



Classifier

"Cat"

$\nabla_{x_t} \log p_\phi(y|x_t)$

X100

t=100

Time Representation

$\epsilon_\theta(x_t, t)$

ε100

$\hat{\epsilon}_\theta(x_t, t)$

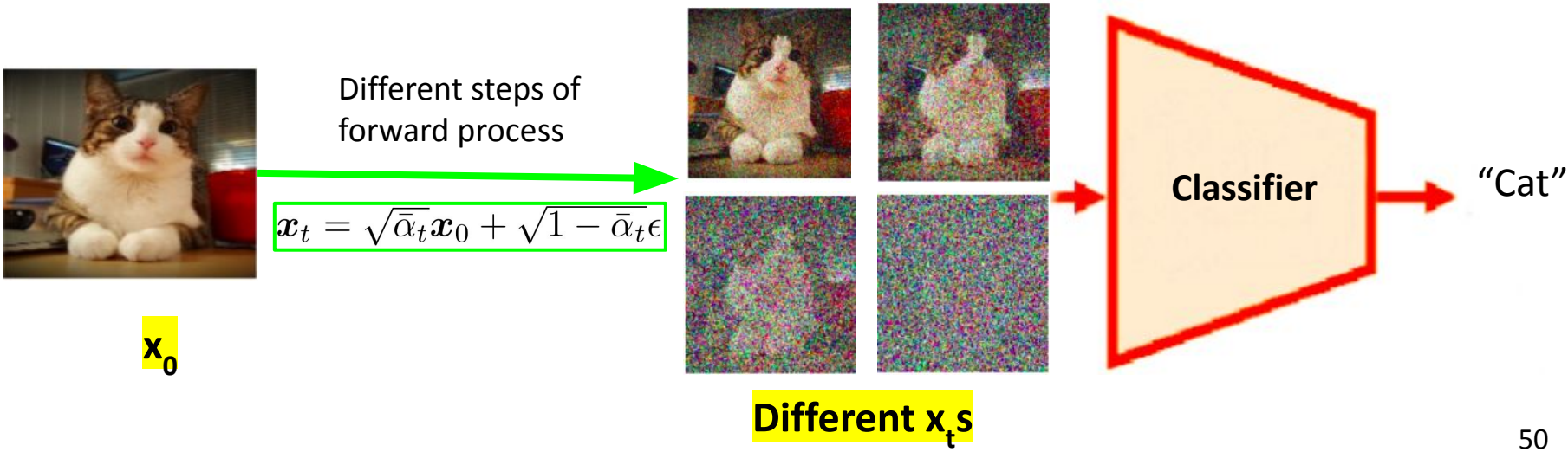
Usual DPM FREEZED!

A network like UNET

$$\hat{\epsilon}_\theta(x_t, t) = \epsilon_\theta(x_t, t) - \sqrt{1-\bar{\alpha}_t} \cdot \nabla_{x_t} \log p_\phi(y|x_t)$$

How to train such a classifier?

- Suppose we have a **pre-trained fixed DPM**
- First, we train a classifier on the noisy data (add noise based on **fixed DPM hyperparameters**)



Conditional Sampling Algorithm (using DDPM)

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

How can this form obtained?

$$\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \sqrt{1-\bar{\alpha}_t} \cdot \nabla_{\boldsymbol{x}_t} \log p_\phi(\boldsymbol{y}|\boldsymbol{x}_t)$$

$$p(x \mid y) = \frac{p(y \mid x) \times p(x)}{p(y)}$$

$$\log\left(p(x \mid y)\right) = \log\left(p(y \mid x)\right) + \log\left(p(x)\right) - \log\left(p(y)\right)$$

$$\nabla_x(\log\left(p(x \mid y)\right)) = \nabla_x(\log\left(p(y \mid x)\right)) + \nabla_x(\log\left(p(x)\right)) - \nabla_x(\log\left(p(y)\right))$$

$$\underbrace{\nabla_x(\log\left(p(x \mid y)\right))}_{-\epsilon'} = \underbrace{\nabla_x(\log\left(p(y \mid x)\right))}_{\textit{Conditional term}} + \underbrace{\nabla_x(\log\left(p(x)\right))}_{-\epsilon}$$

Thanks for you Attention!