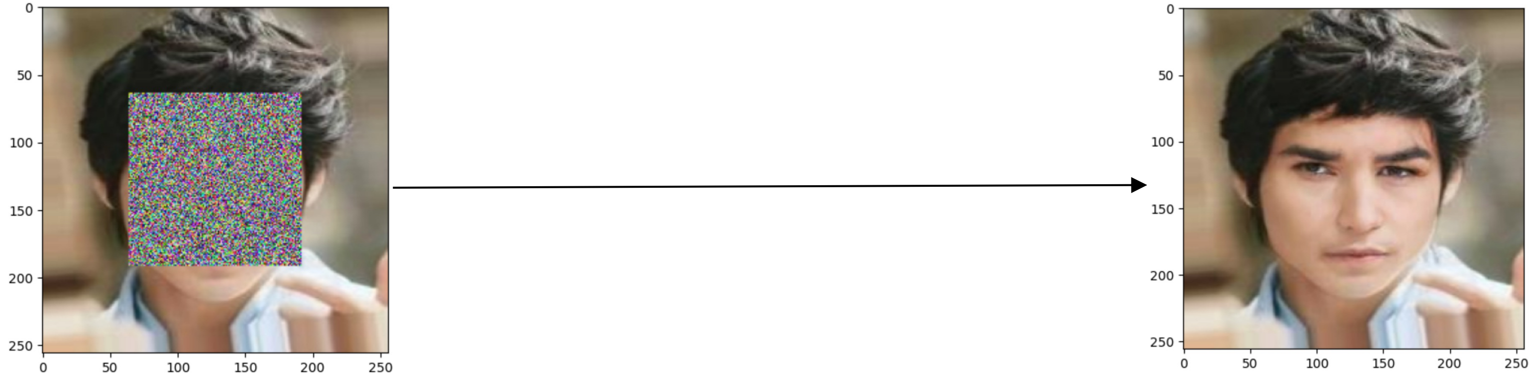
Module 9 - Exercise

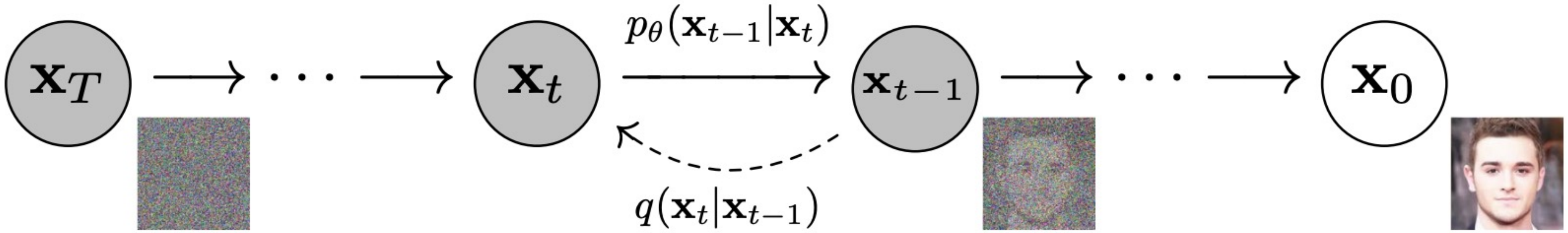
Image Inpainting Using DDPMs

AI VIET NAM
Nguyen Quoc Thai

Objectives

! Image Inpainting using Denoising Diffusion Probabilistic Model





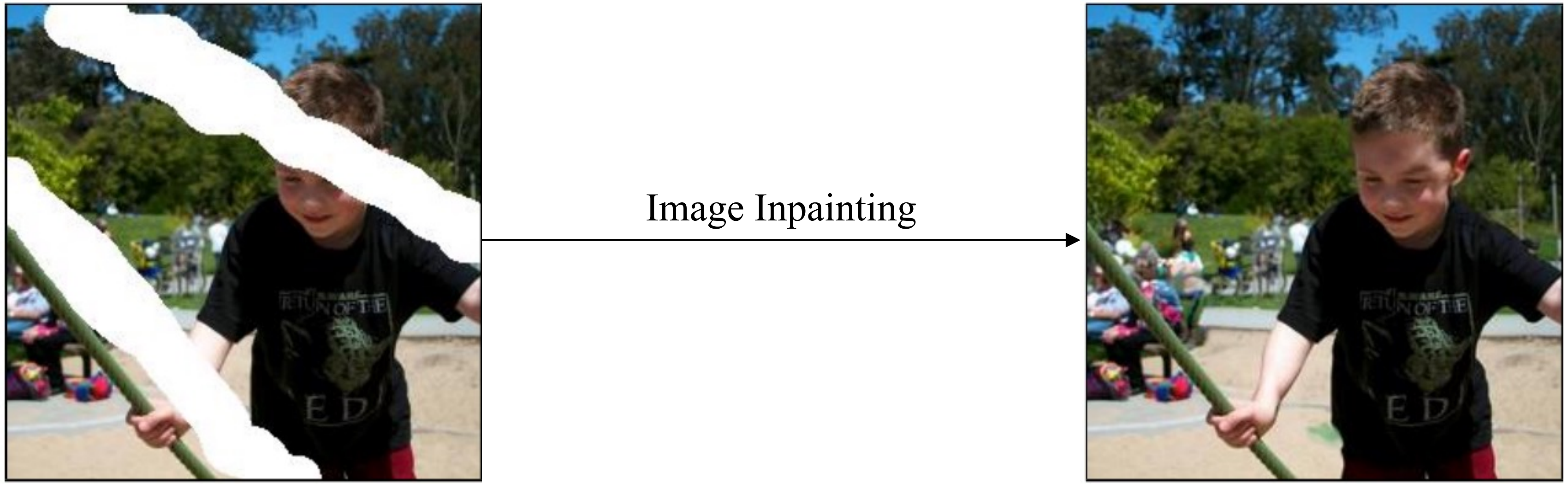
Outline

- Introduction
- Denoising Diffusion Probabilistic Model
- Image Inpainting using DDPM

Introduction

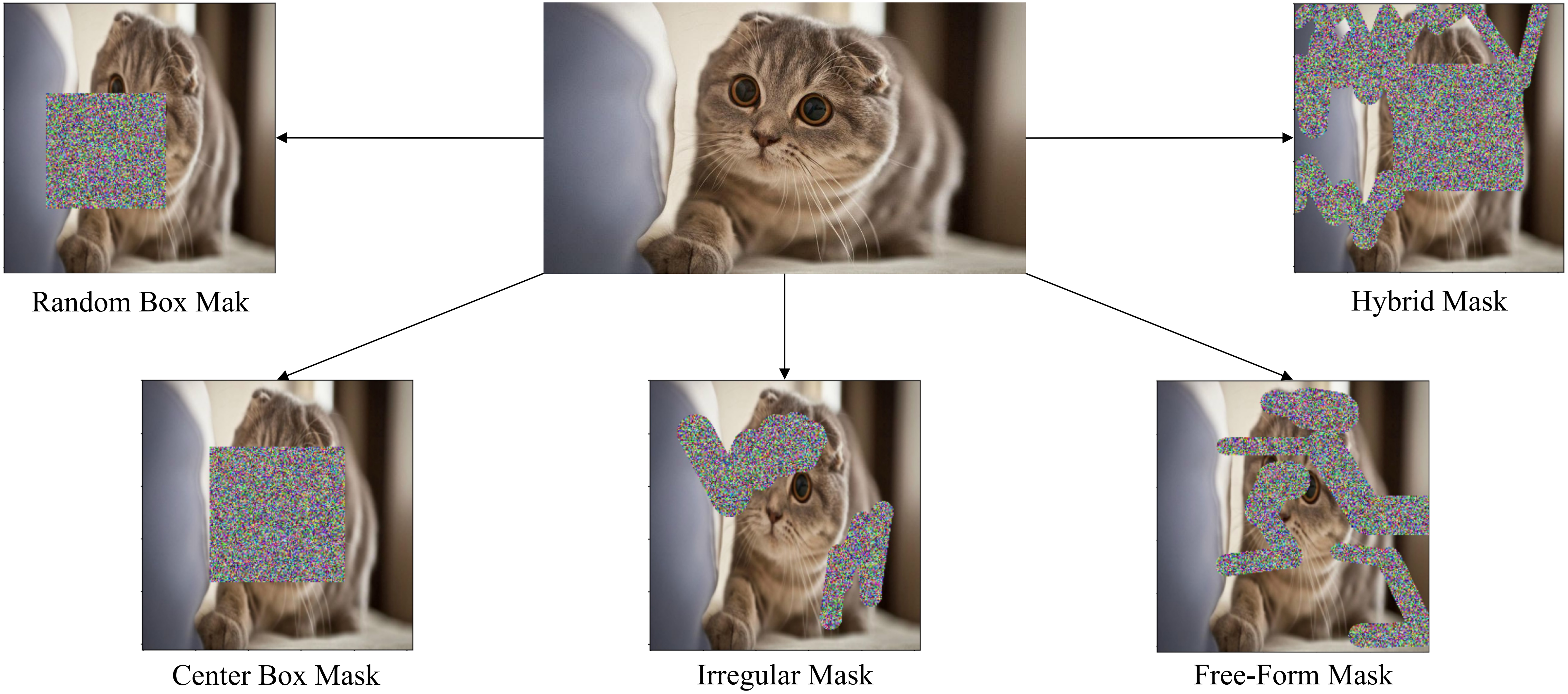
What is Image Inpainting?

- Image Inpainting is to fill missing parts of image
- Important Point: Fill in the hole with
  - Visually realistic
  - Semantically plausible



Introduction

Different Masking Techniques



Introduction

Two main approach

- Patch Pasting
  - Search similar patches to hole-surrounding areas
  - PatchMatch, Scene Completion,…
- Deep Learning Methods
  - Learn how to fill the hole using large scale dataset
  - GLCIC, Gated Conv, Generative Models,…

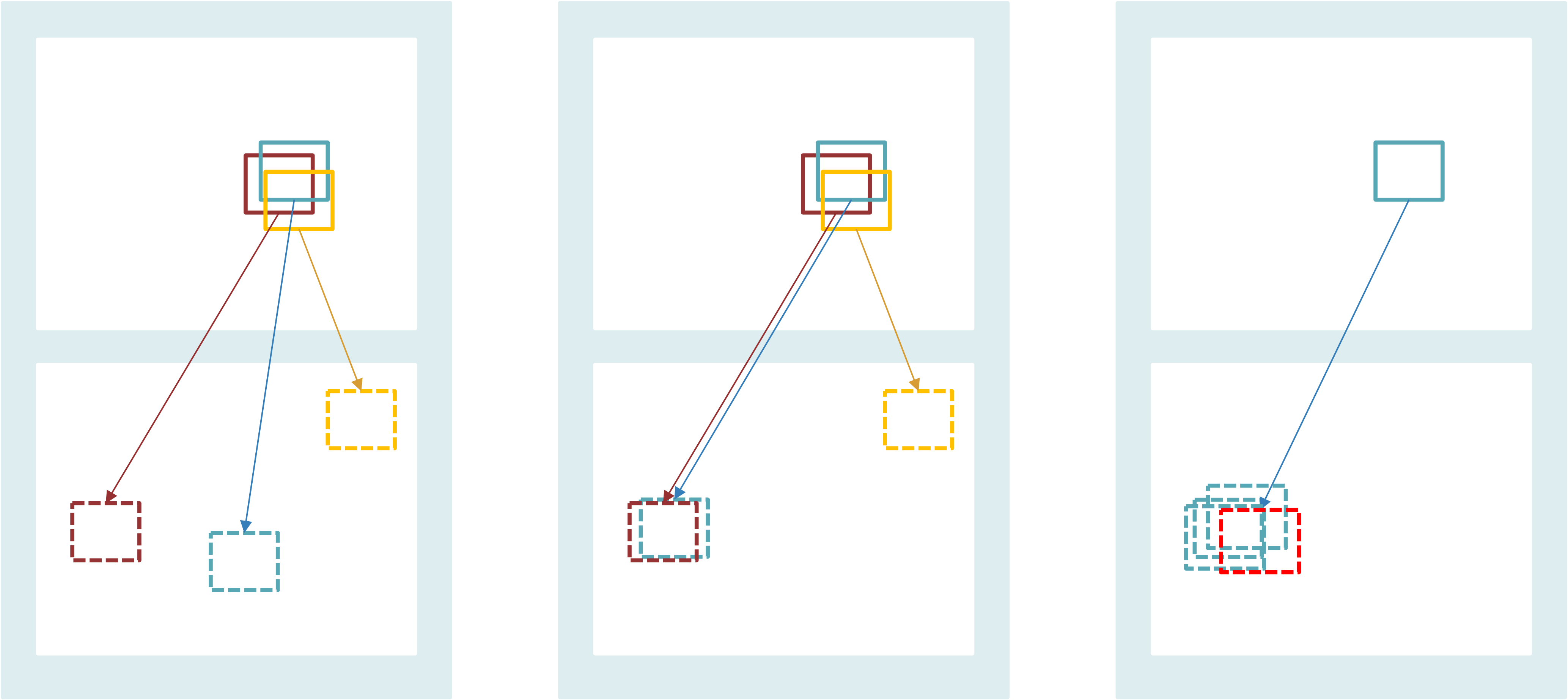
Introduction

PatchMatch

- Search similar patch to the hole surrounding areas in an input image
- Paste the searched patch to the hole
- Loop

Introduction

PatchMatch



Initialization Propagation Search

Introduction

Scene Completion

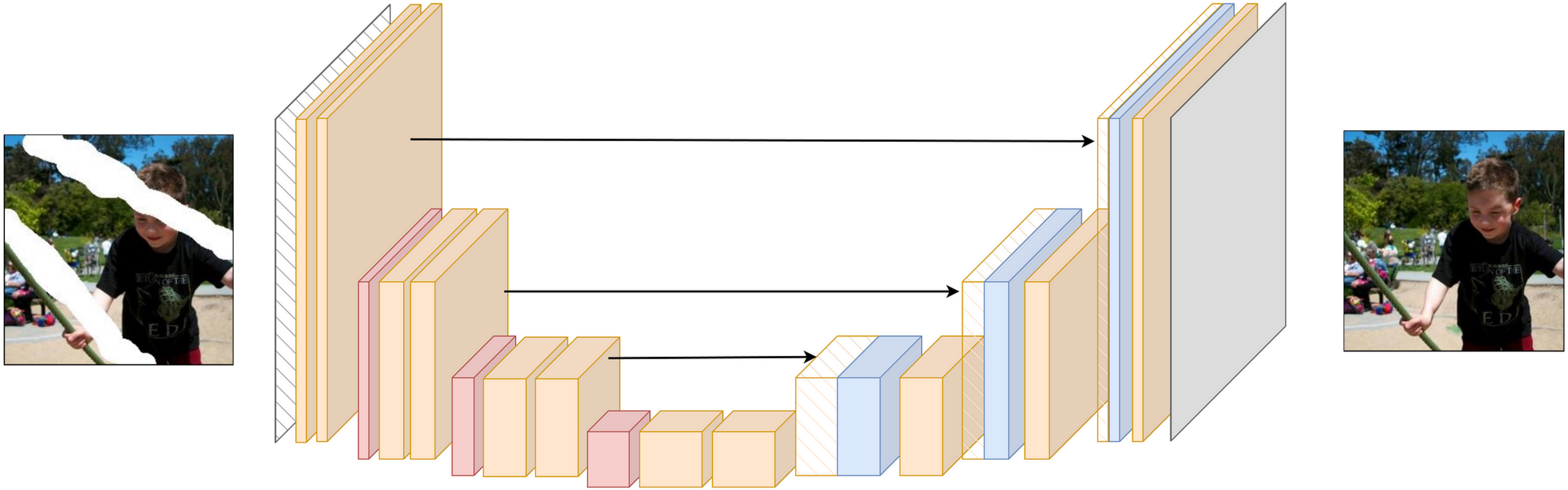
- Search the image similar to an input from large scale image database
- Search similar patch and paste it to the input's hole

Original Image Input Scene Matches Output

Introduction

Deep Learning Methods

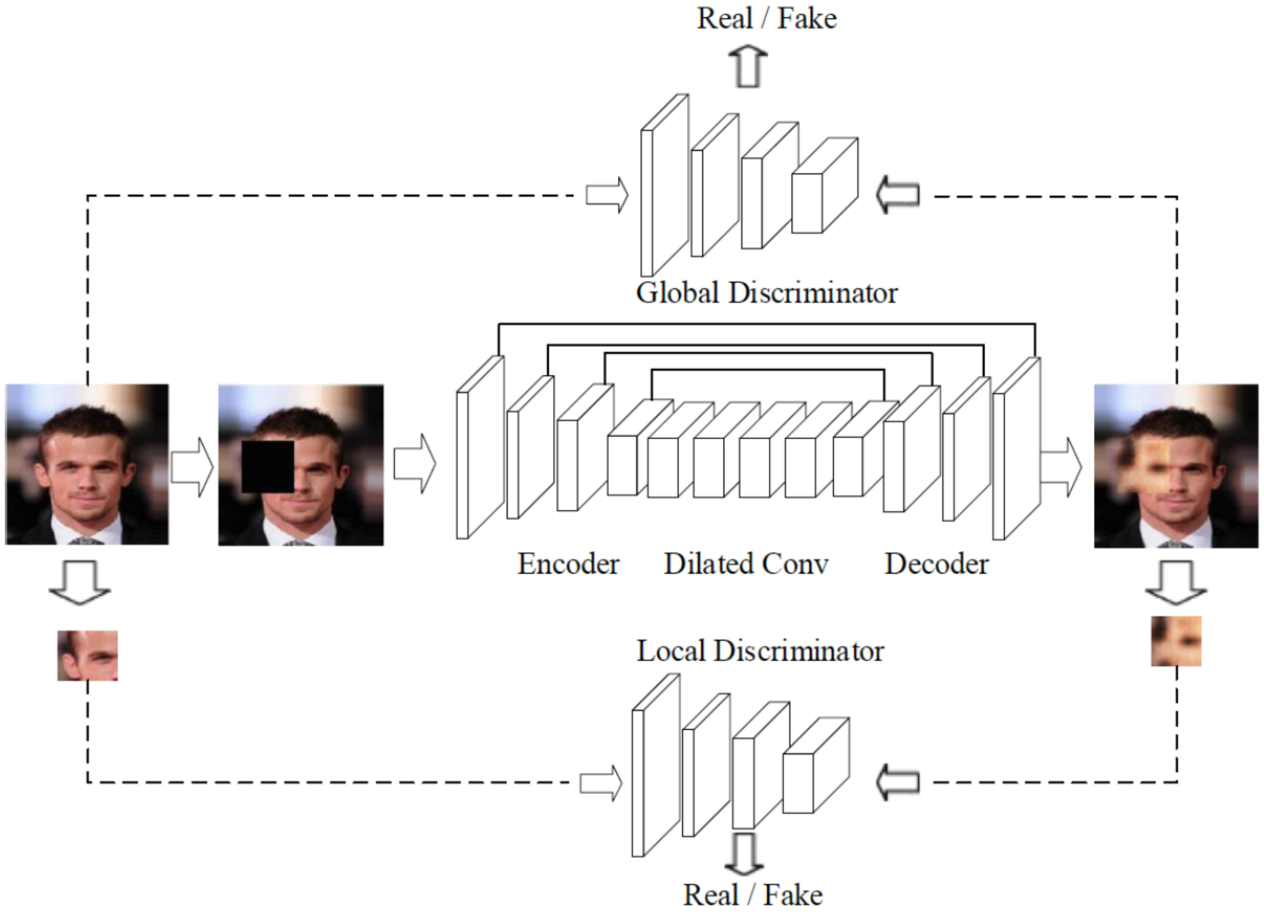
- Generate the contents of the hole using CNNs from the large dataset
- Encoder-Decoder CNN



Introduction

Deep Learning Methods

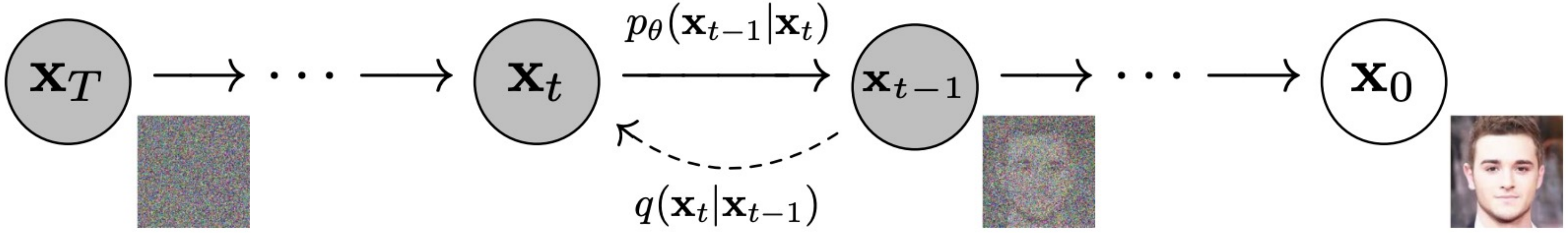
- Global and Locally Consistent Image Completion

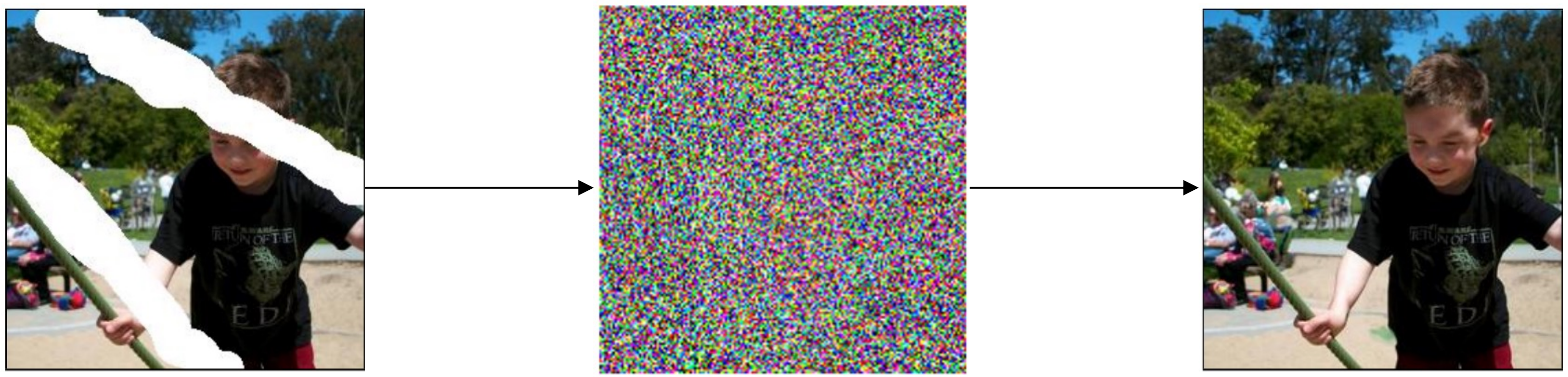


Introduction

Deep Learning Methods

- Diffusion Models





Outline

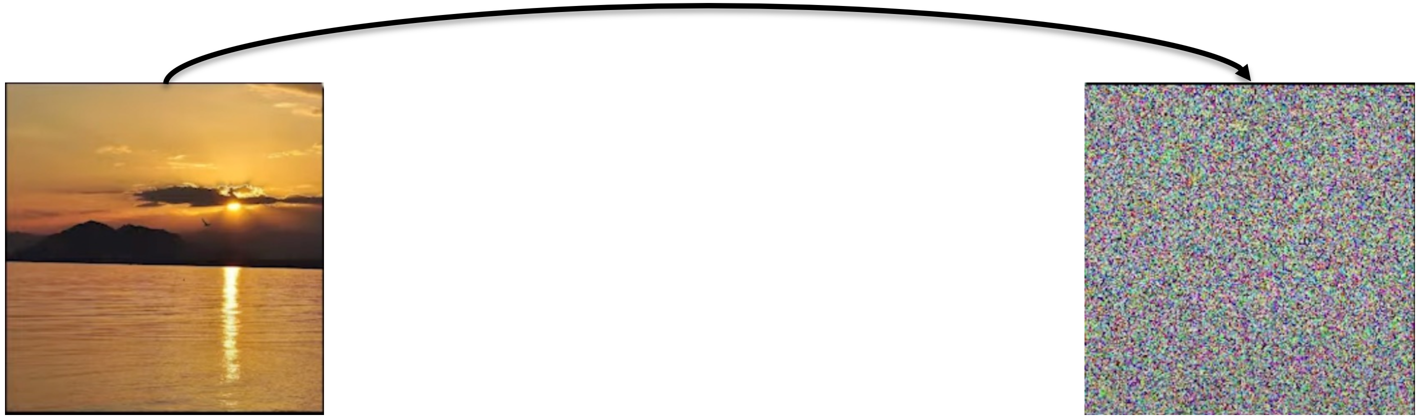
- Introduction
- Denoising Diffusion Probabilistic Model
- Image Inpainting using DDPM

AI VIET NAM
@aivietnam.edu.vn

DDPMs

Objectives

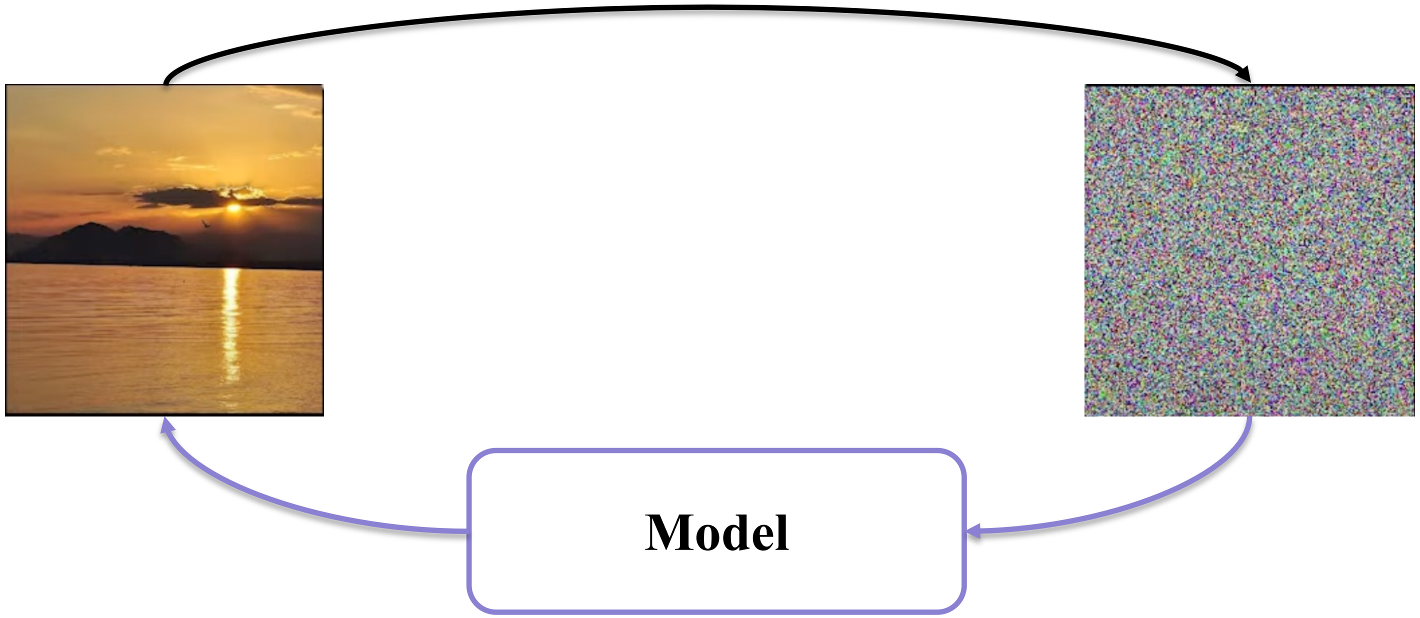
➢ Destroy structure in a data distribution through an iterative **forward diffusion process**



DDPMs

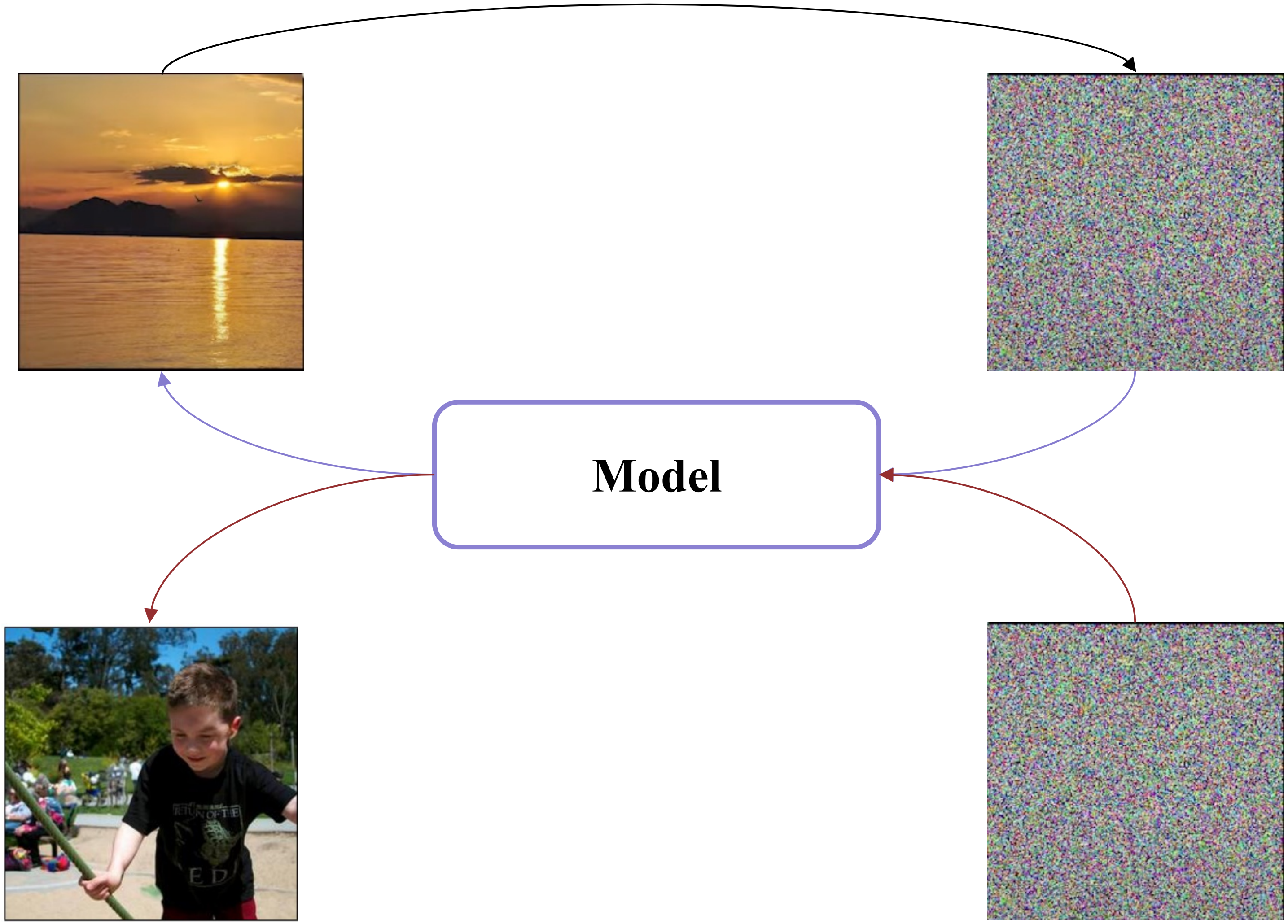
Objectives

- Destroy structure in a data distribution through an iterative **forward diffusion process**
- Learn a **reverse diffusion process** that restores structure in data



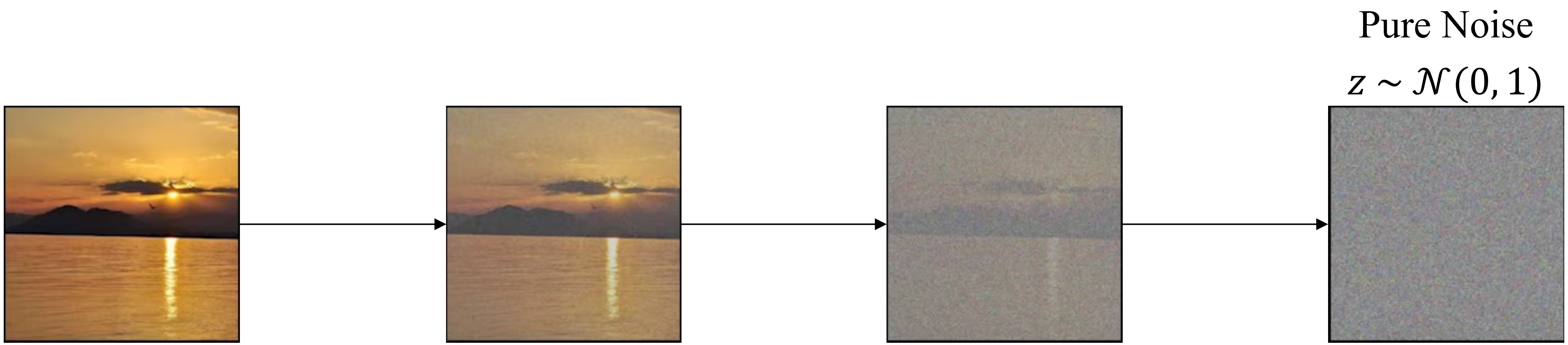
DDPMs

Objectives



DDPMs

Forward Diffusion Process



DDPMs

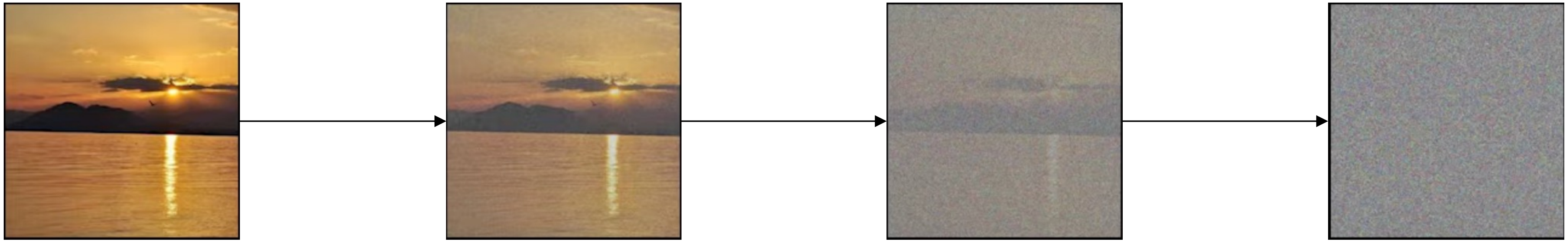
Forward Diffusion Process

Regulated by a schedule

β

Pure Noise

$z \sim \mathcal{N}(0, 1)$



x_0 x_{100} x_{200} x_{300}

T=300 (timestep)

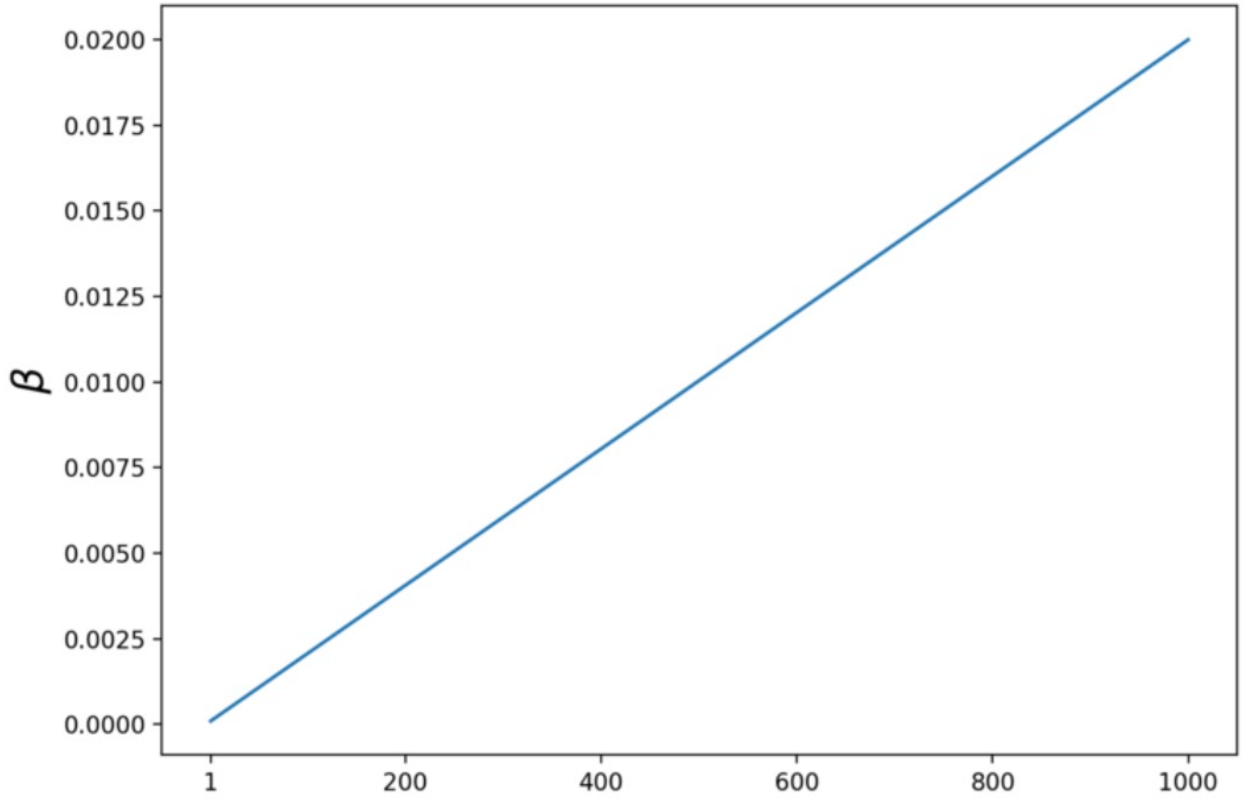
DDPMs

Forward Diffusion Process

- Linear Noise Schedule

$\beta_{end} = 0.02$

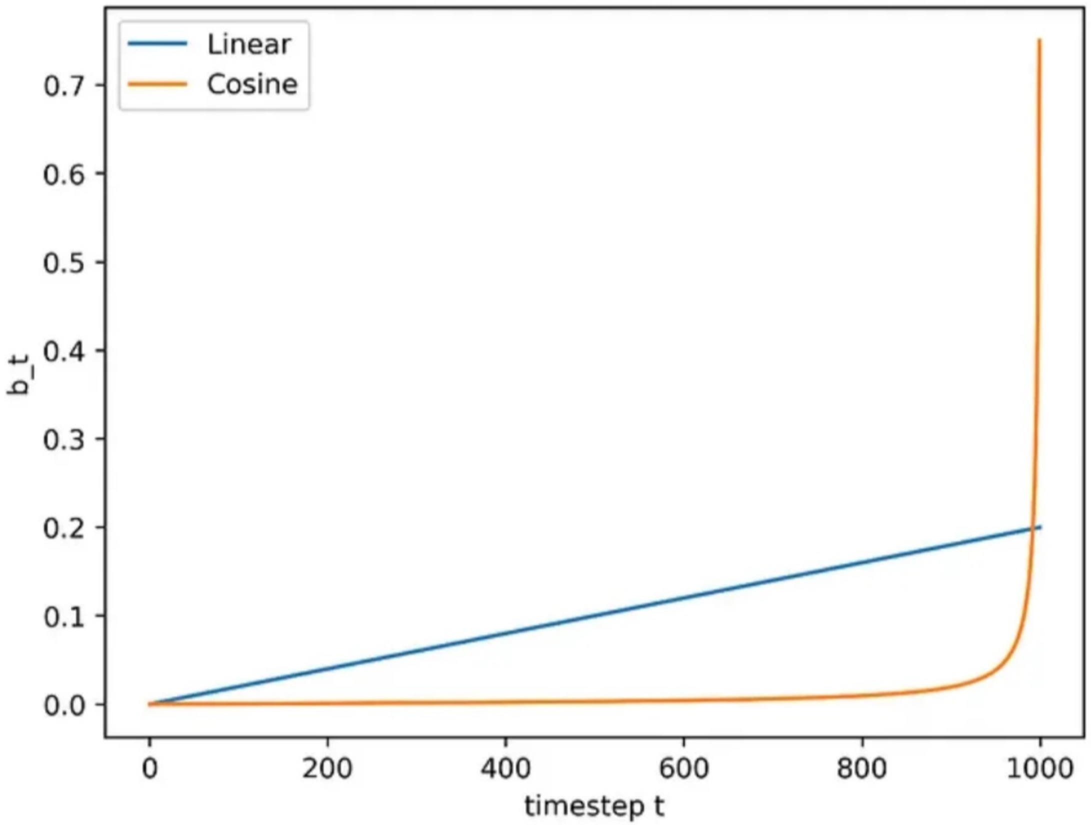
$\beta_{start} = 0.0001$

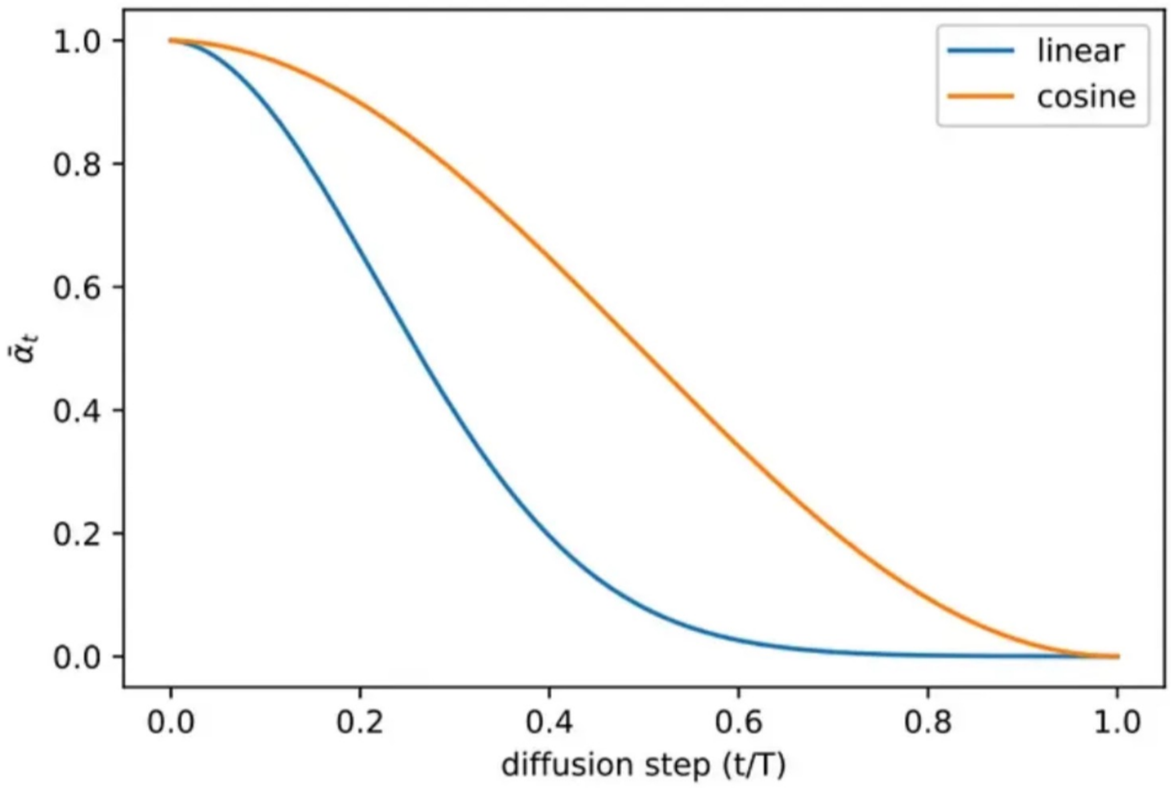


DDPMs

Forward Diffusion Process

- Noise Schedule



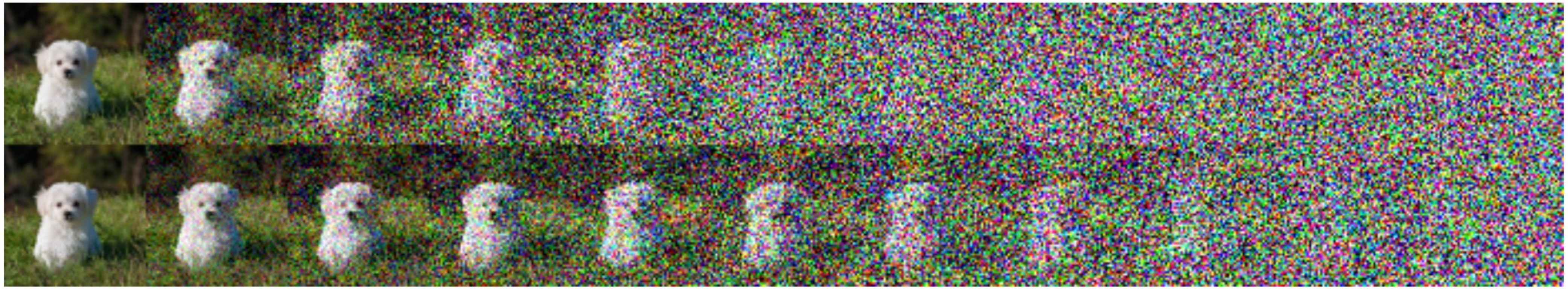


DDPMs

Forward Diffusion Process

- Noise Schedule

Linear Schedule



Cosine Schedule

DDPMs

Forward Diffusion Process

$$q(x_t|x_{t-1})$$

$$q(x_t|x_{t-1}) = \overset{\text{Normal distribution}}{\mathcal{N}}\Big(\underset{\text{Output}}{x_t}; \overset{\text{Mean}}{\sqrt{1-\beta_t}\,x_{t-1}}, \underset{\text{Variance}}{\beta_t I}\Big)$$

DDPMs

Forward Diffusion Process

$$q(x_t|x_{t-1})$$

$$\alpha_t = 1 - \beta_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} a_s$$

$$\mathcal{N}(\mu, \sigma^2) = \mu + \sigma \cdot \epsilon$$

$$\begin{aligned} q(x_t|x_{t-1}) &= \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right) \\ &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \\ &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \end{aligned}$$

DDPMs

Forward Diffusion Process

$$q(x_t|x_{t-1})$$

x_0 x_{t-1} x_t x_T

$$\alpha_t = 1 - \beta_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t} a_s$$

$$\mathcal{N}(\mu, \sigma^2) = \mu + \sigma \cdot \epsilon$$

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$
$$= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}I\right)$$

DDPMs

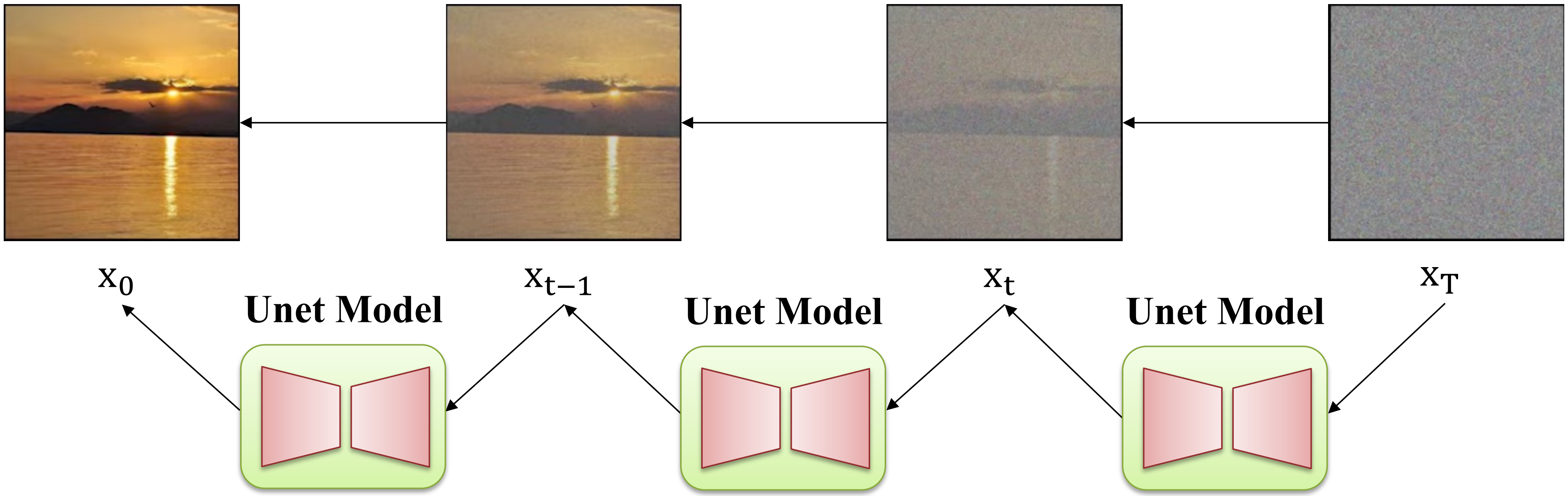
Reverse Diffusion Process

$$q(x_{t-1}|x_t, x_0)$$

DDPMs

Reverse Diffusion Process

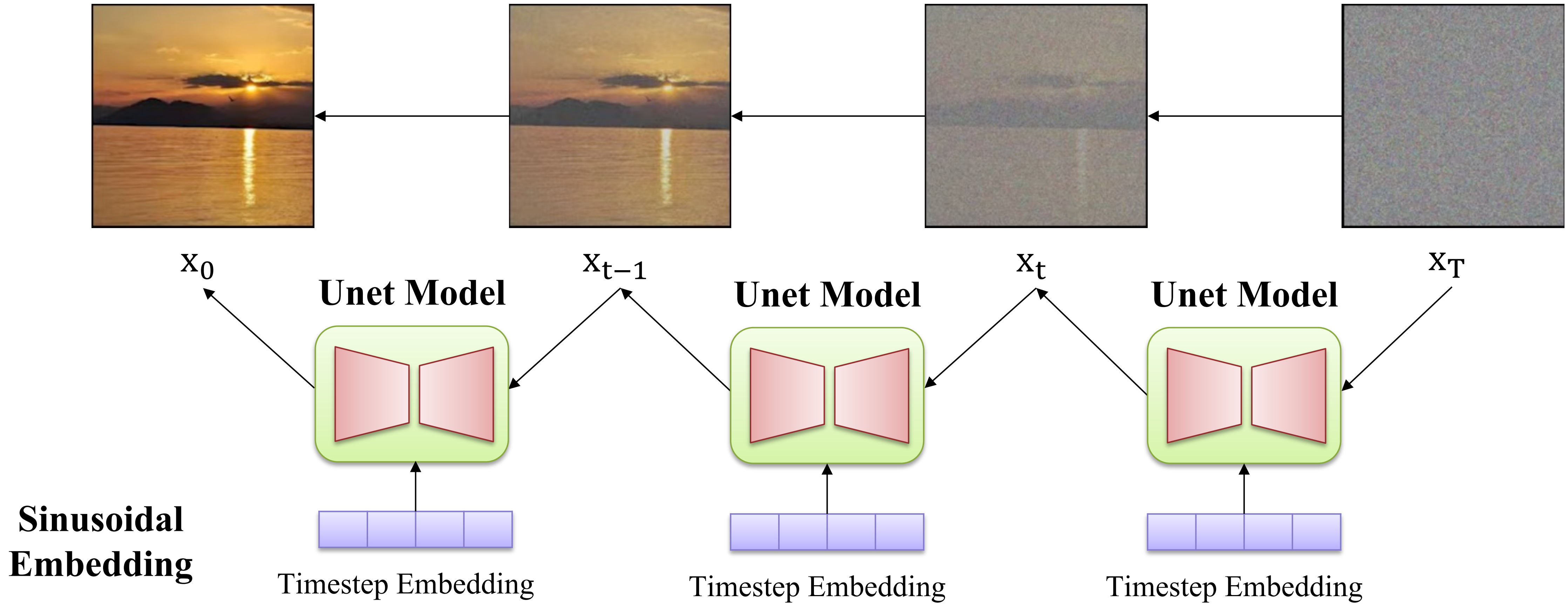
$$q(x_{t-1}|x_t, x_0)$$



DDPMs

Reverse Diffusion Process

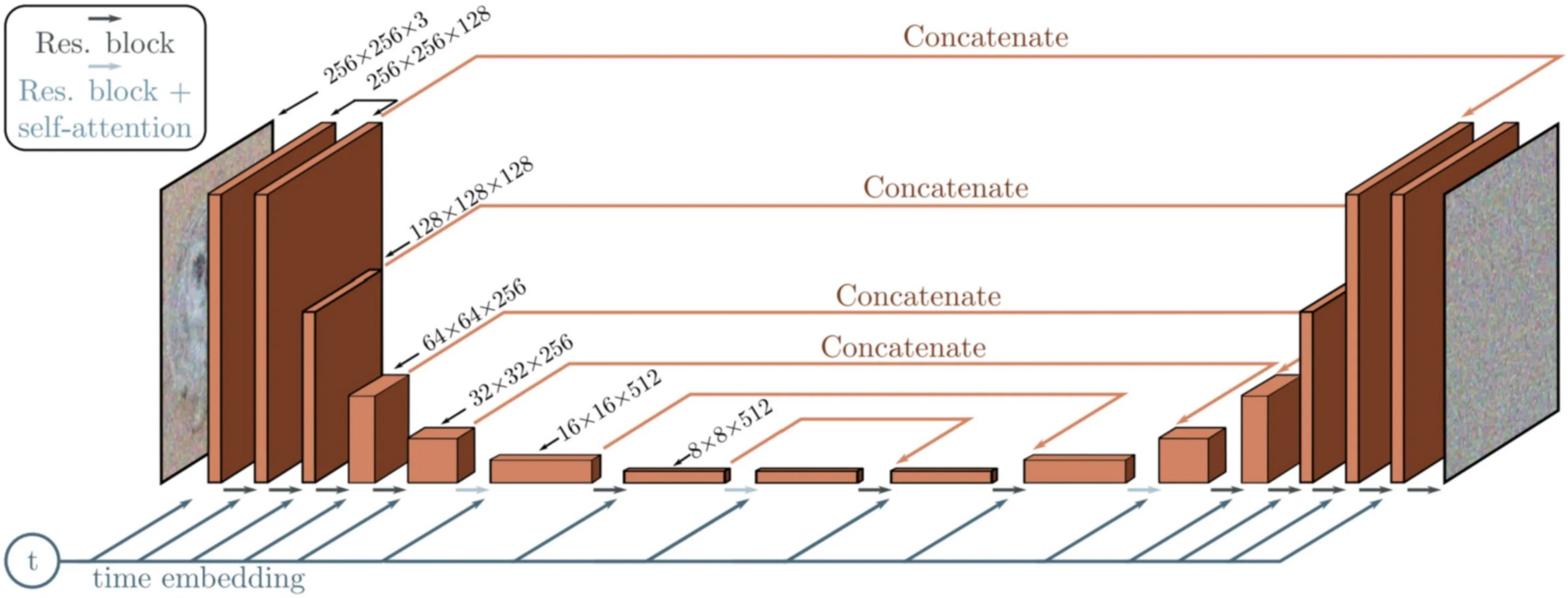
$$q(x_{t-1}|x_t, x_0)$$



DDPMs

Reverse Diffusion Process

- UNet Model with Self-Attention

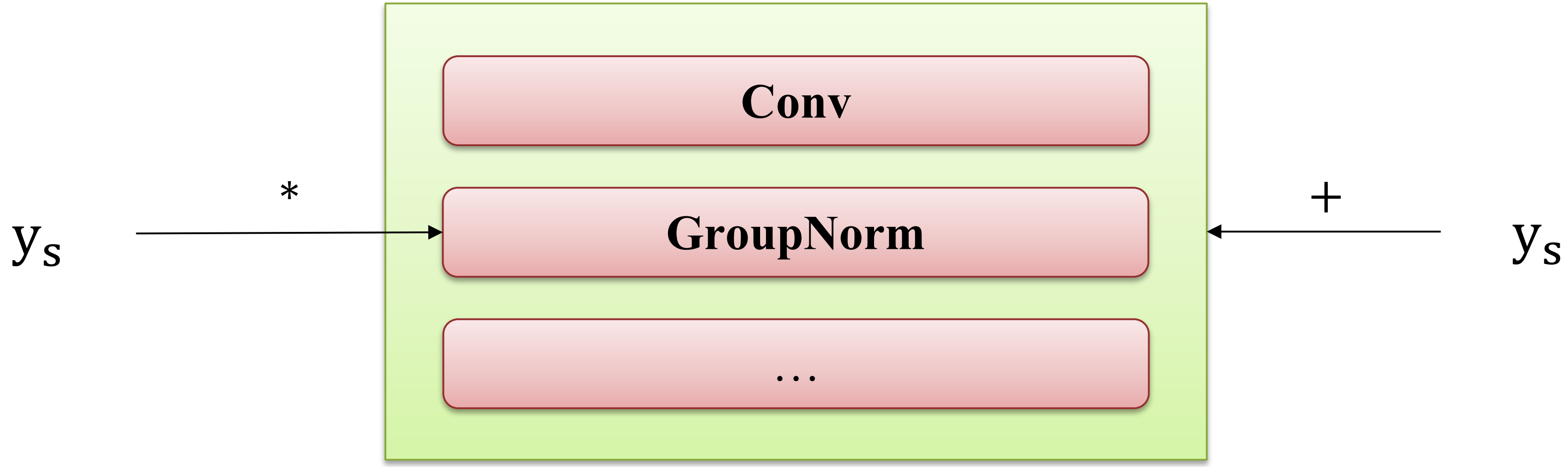


DDPMs

Reverse Diffusion Process

- UNet Model with Self-Attention
- Adaptive Group Normalization

$$\mathrm{AdaGN}(h, y = y_s \mathrm{GroupNorm}(h + y_b)$$



DDPMs

Reverse Diffusion Process

$$\mathbf{q}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\boldsymbol{\beta}}_t\right)$$

$$\tilde{\mu}(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$$

$$\tilde{\beta}_t = \frac{(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\beta_t$$

DDPMs

Training

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

6: **until** converged

DDPMs

Sampling

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 $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

Outline

- Introduction
- Denoising Diffusion Probabilistic Model
- Image Inpainting using DDPM

Image Painting using DDPMs

CelebA Dataset

Image Painting using DDPMs

Center Box Mask

Masking

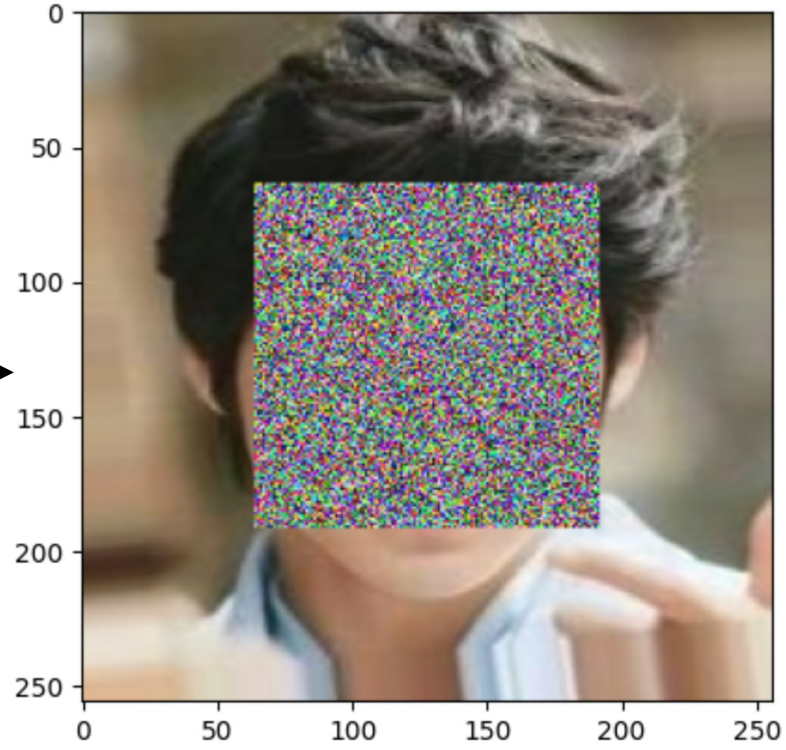


Image Painting using DDPMs

Approach

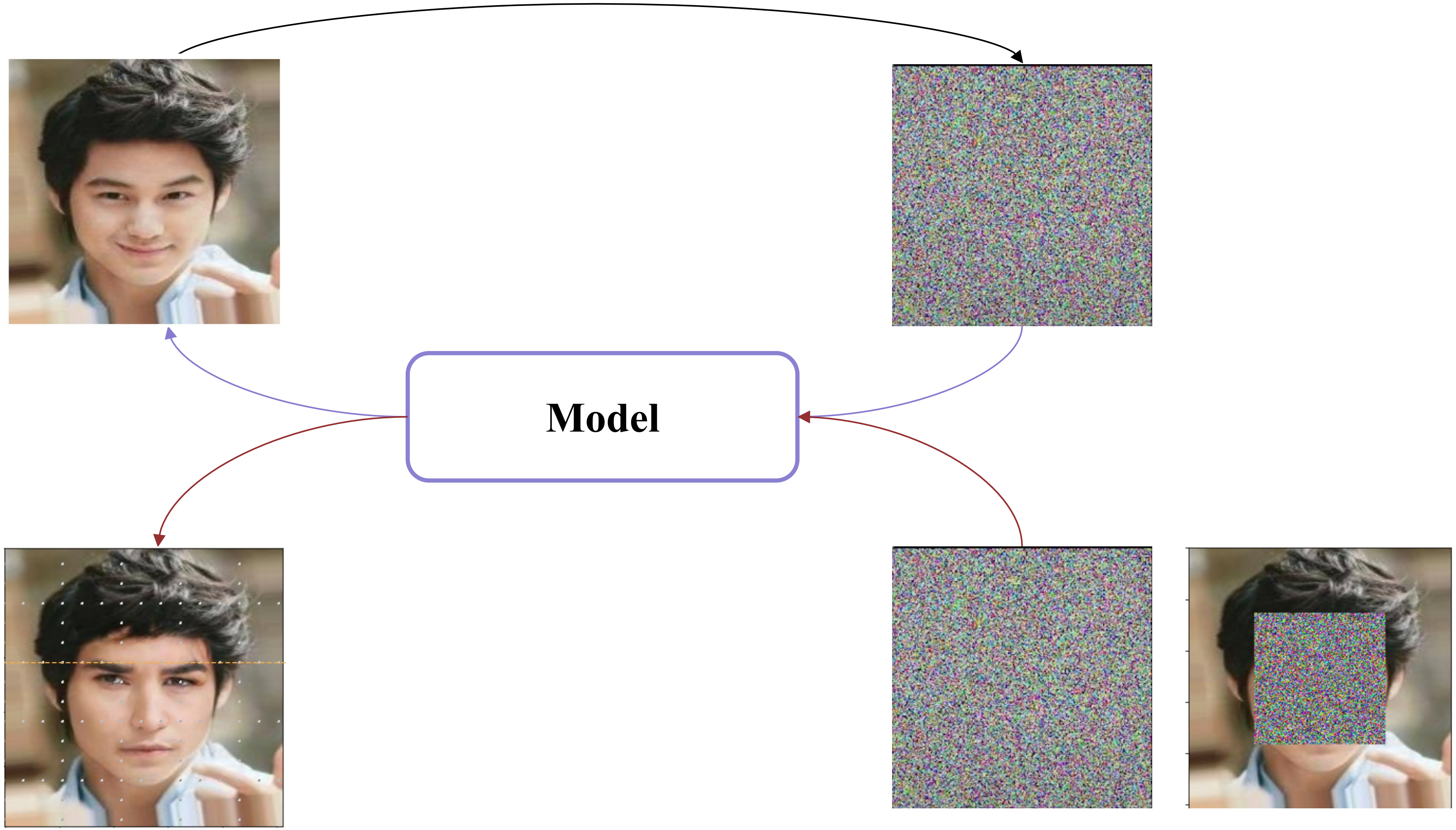


Image Painting using DDPMs

Reverse Diffusion Model

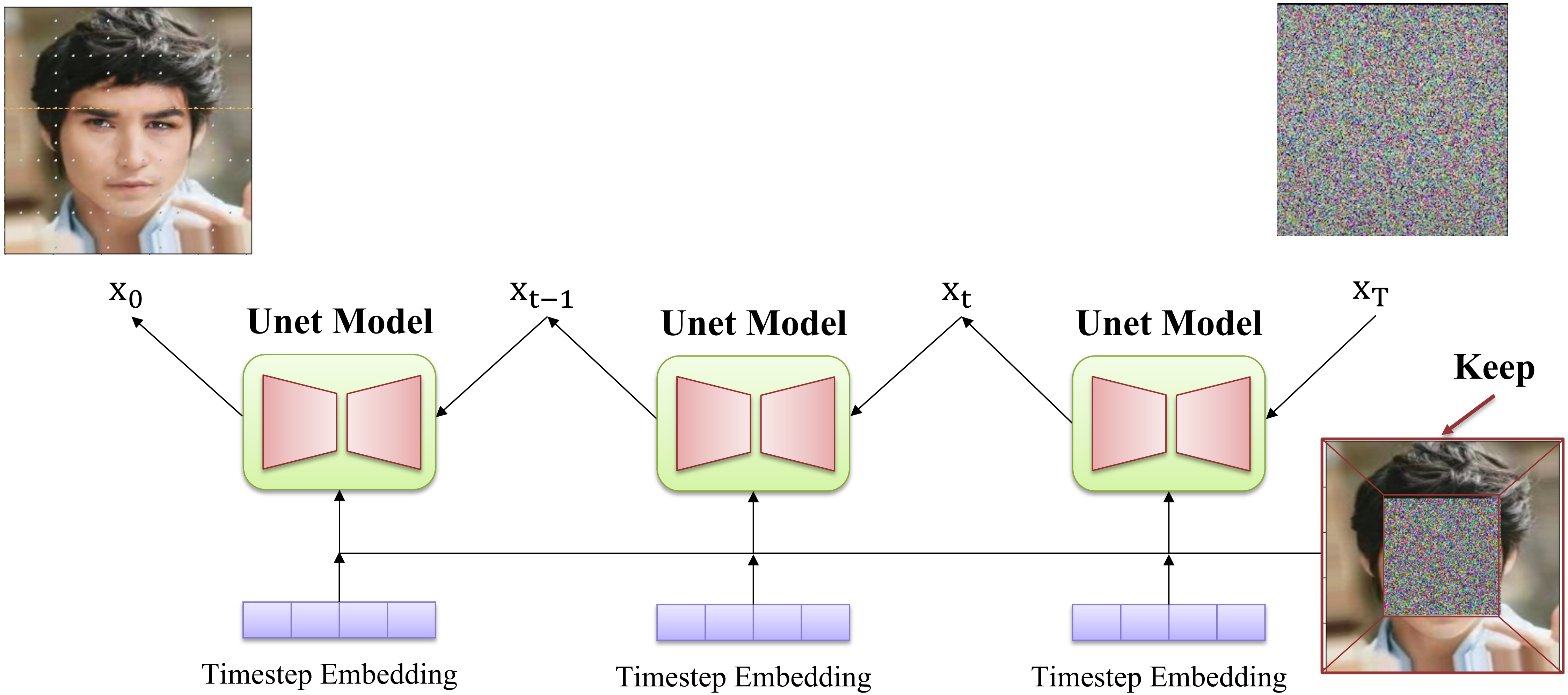
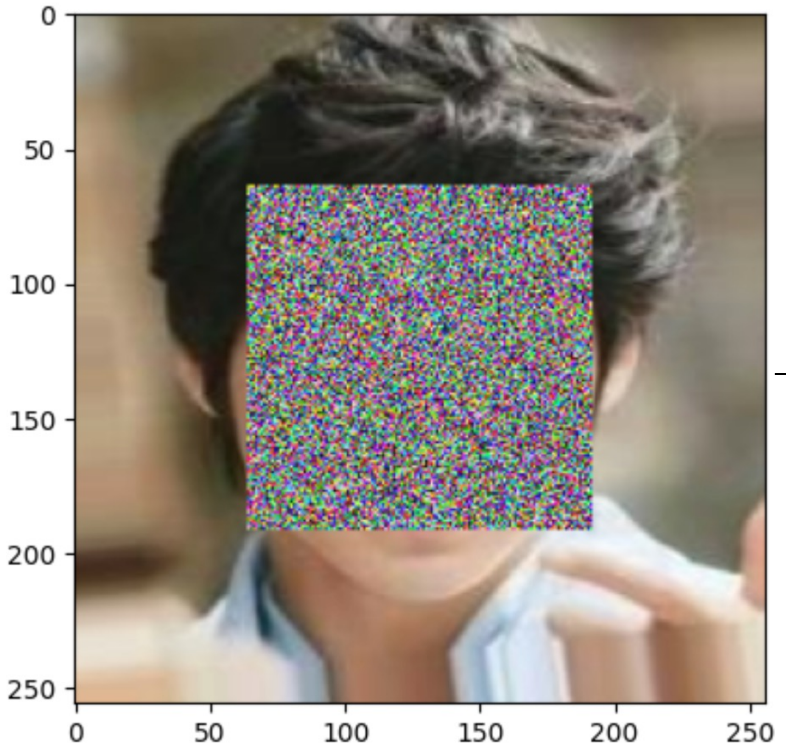
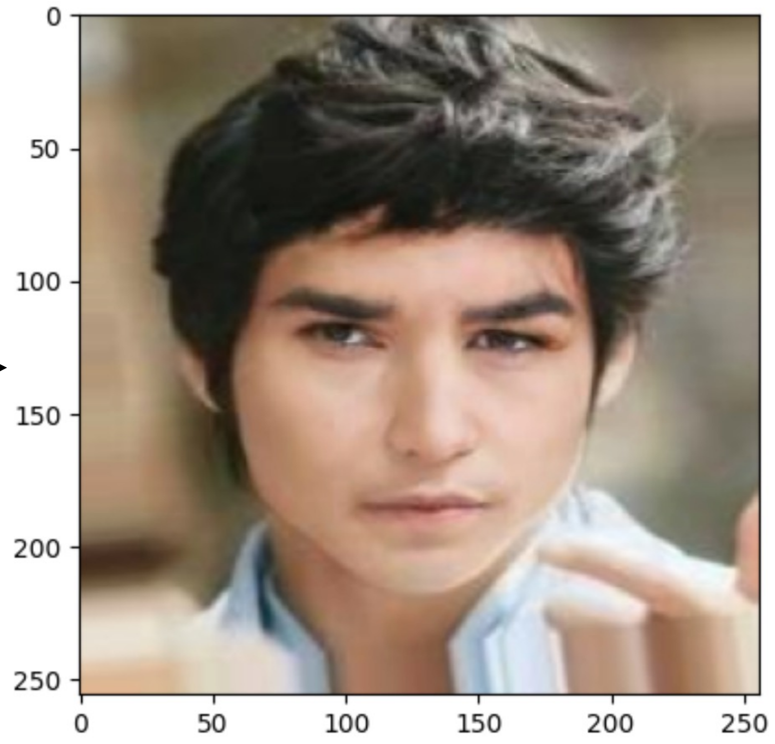


Image Painting using DDPMs

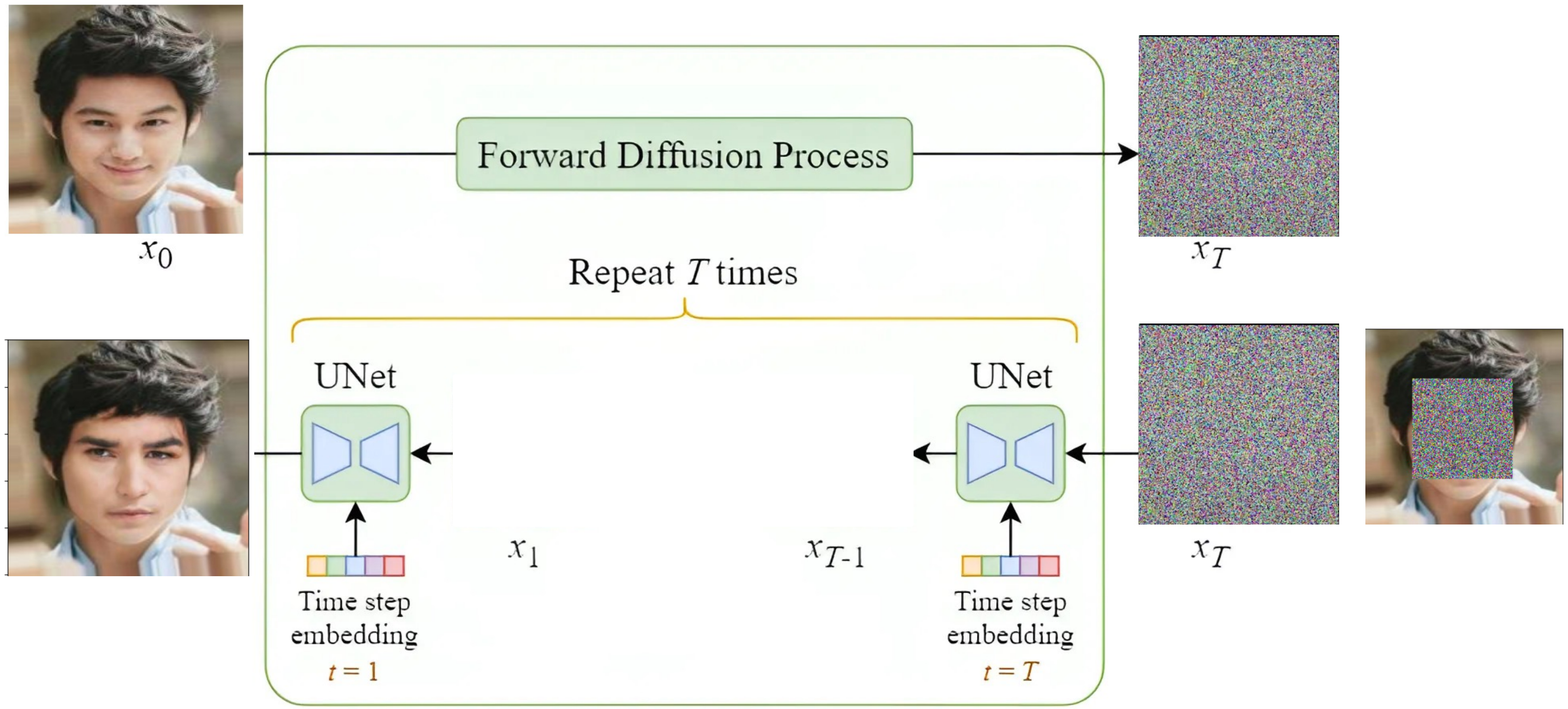
Experiment



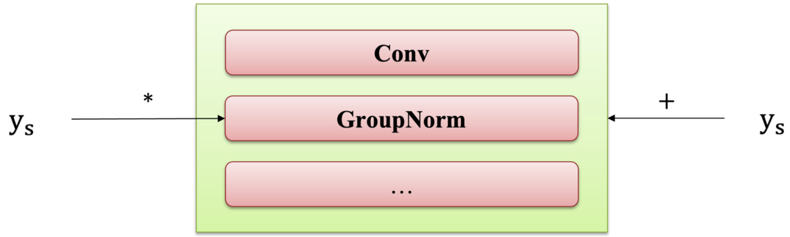
Inpainting

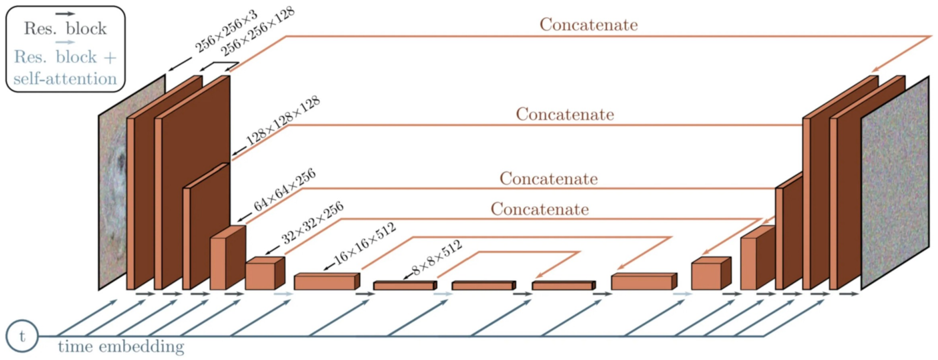


Summary



$\text{AdaGN}(h, y = y_s \text{GroupNorm}(h + y_b)$





Thanks!

Any questions?