Lecture 21: Diffusion Models
CS480/680 Intro to Machine Learning

2023-3-28

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

Preview

```
seed_and_prompt_sequence = [
        3764, 'in the beginning there was nothing, just darkness',
        1537, 'special effects render of the big bang',
        6573, 'HD photo of a large amount of spiral galaxies',
        1791, 'early planet formation in the solar system',
        9973, 'the Hadean earth was bombarded with asteroids and massive volcanic eruptions',
        736,  'panoramic view of earth with ocean surrounding newly formed land and volcanos',
        3639, 'hydrothermal vents at the bottom of the ocean',
        3559, 'bacteria under a microscope',
        4724, 'bacteria under a microscope',
        3359, 'ammonites floating in the ocean',
        6344, 'the first reptile to leave the ocean and crawl onto the land',
        6344, 'the first reptile to leave the ocean and crawl onto the land',
        6813, 'massive brachiosaurus walking amidst a green mountain range',
        6678, 'the exctinction of the dinosaurs be a huge meteorite',
        7450, 'small mammals thriving in a cave',
        9766, 'small, prehistoric mammals living in the jungle',
        5009, 'group of monkeys in the forest',
        7287, 'HD photograph of neanderthal, the first man',
        6008, 'cave painting',
        208,  'cavemen tribe gathered around a fire at night looking at the stars',
        2222, 'maasai tribe hunting on the savanna with spears',
        571,  'homo sapiens using stone tools',
        632,  'a small, tribal village with huts',
        1332, 'at the dawn of civilization, small villages emerged',
        2496, 'ancient egypt, the first massive civiliation',
        1869, 'the height of the roman empire, incredible architecture, by Greg Rutkowski',
        7559, 'medieval town square',
        1265, 'medieval city',
        6628, 'the skyline of New York city',
```

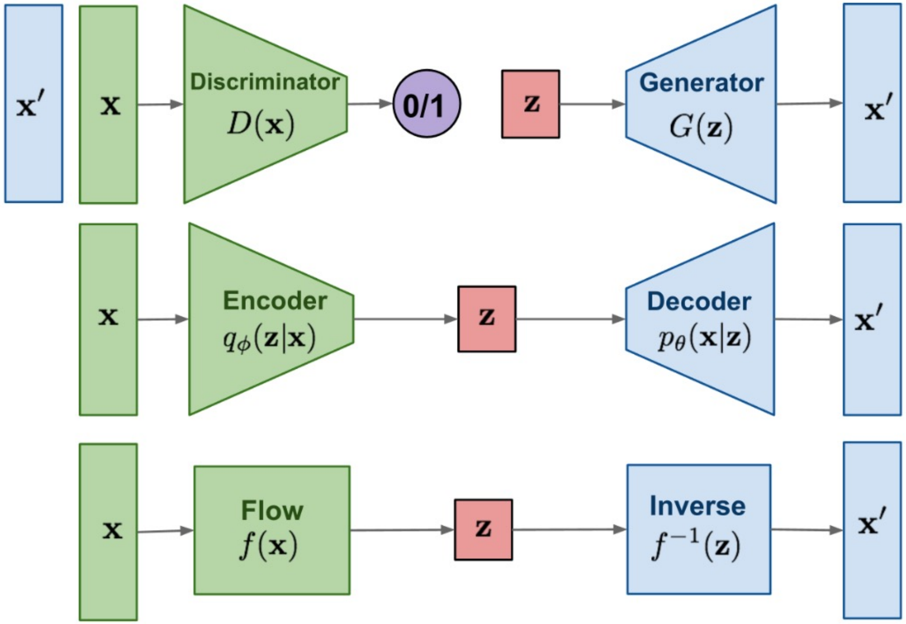
Xander Steenbrugge created the amazing **Voyage through Time** video below using stable diffusion with the input prompts shown in the figure.

Recap

GAN: Adversarial training

VAE: maximize variational lower bound

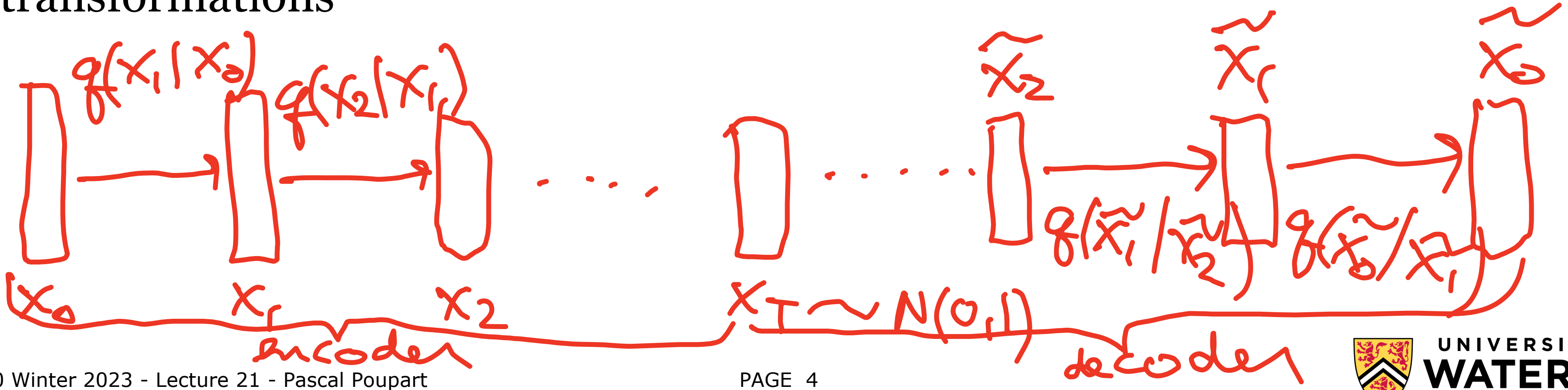
Flow-based models: Invertible transform of distributions



From https://lilianweng.github.io/

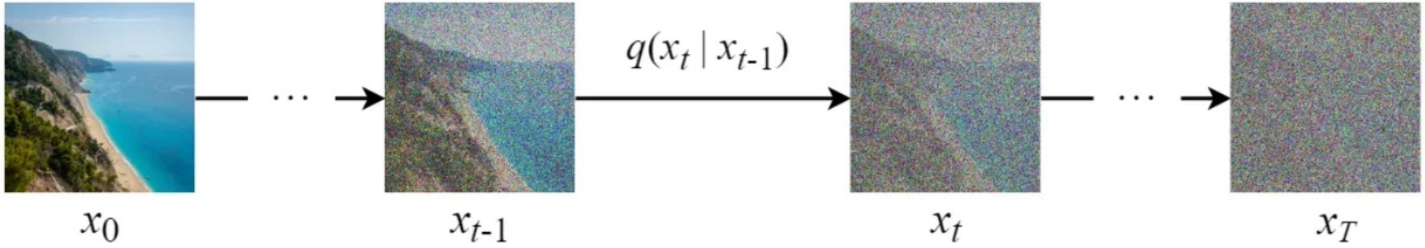
Diffusion Model

- Stochastic autoencoder (encoder introduces noise and decoder denoises the data)
  - Generates better data than variational autoencoders
  - Easier to train than generative adversarial networks and does not suffer from mode collapse
  - Special type of stochastic flow that is not restricted to invertible transformations



UNIVERSITY OF WATERLOO

Forward Diffusion Process



Distribution of the noised images — Output — Mean μ_t — Variance Σ_t

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Notations:

t : time step (from 0 to T)

x_0 : a data sampled from the real data distrition $q(x)$ (i.e. $x_0 \sim q(x)$)

β_t : variance schedule ($0 \le \beta_t \le 1$, and β_0 = small number, β_T = large number)

I : identity matrix

From Steins (medium.com)

Stochastic Transformation

- Recall the reparameterization trick:
  - When $x \sim P(x) = N(x|\mu, \sigma^2)$
    then $x = \sigma\epsilon + \mu$ where $\epsilon \sim N(\epsilon|0,1)$

- Since $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = N\left(\boldsymbol{x}_t\middle|\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}\right)$
  Then $\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$

Stochastic Transformation

- We can speed up the noise process by computing x_t in one step:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}|\boldsymbol{0},\boldsymbol{1})$

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

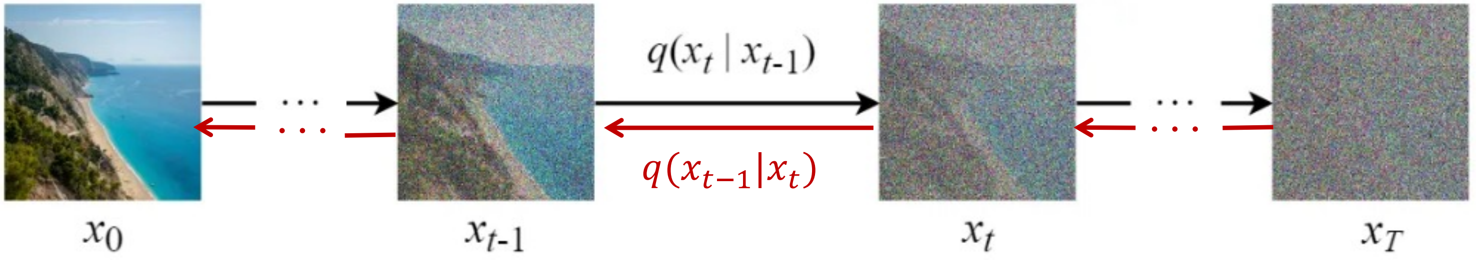
See https://medium.com/@steinsfu/diffusion-model-clearly-explained-cd331bd41166 for derivation

- In the limit, $\boldsymbol{x}_\infty$ is a random vector from an isotropic Gaussian

$$\lim_{t\to\infty} \boldsymbol{x}_t = \sqrt{\bar{\alpha}_\infty}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_\infty}\boldsymbol{\epsilon} = \boldsymbol{\epsilon} \;\text{ since } \bar{\alpha}_\infty \to 0$$

Reverse Denoising Process

From Steins (medium.com)



- Forward factorization: $q(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = q(\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$
- Reverse factorization: $q(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = \prod_{t=1}^{T} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\, q(\boldsymbol{x}_T)$
  - Since joint distribution is Gaussian then $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is also Gaussian
  - $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(\boldsymbol{x}_{t-1}|\tilde{\mu}_t(\boldsymbol{x}_t, t), \sigma_t \boldsymbol{I})$

Reverse Conditional Gaussian

- The reverse conditional $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(\boldsymbol{x}_{t-1}|\tilde{\mu}_t(\boldsymbol{x}_t, t), \sigma_t \boldsymbol{I})$ does not have a closed form, but Ho, Jain and Abbeel (2020) derived the following approximation for $\tilde{\mu}_t$:

$$\tilde{\mu}_t(\boldsymbol{x}_t, t) \approx \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$$

where $\boldsymbol{\epsilon}_t = N(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$ is the noise introduced at step t

- We do not know $\boldsymbol{\epsilon}_t$, but we can train a neural network $\epsilon_\theta(\boldsymbol{x}_t, t)$ to approximate it:

$$\text{Minimize } L(\theta) = \left|\left|\boldsymbol{\epsilon}_t - \epsilon_\theta(\boldsymbol{x}_t, t)\right|\right|_2^2$$

Training Algorithm

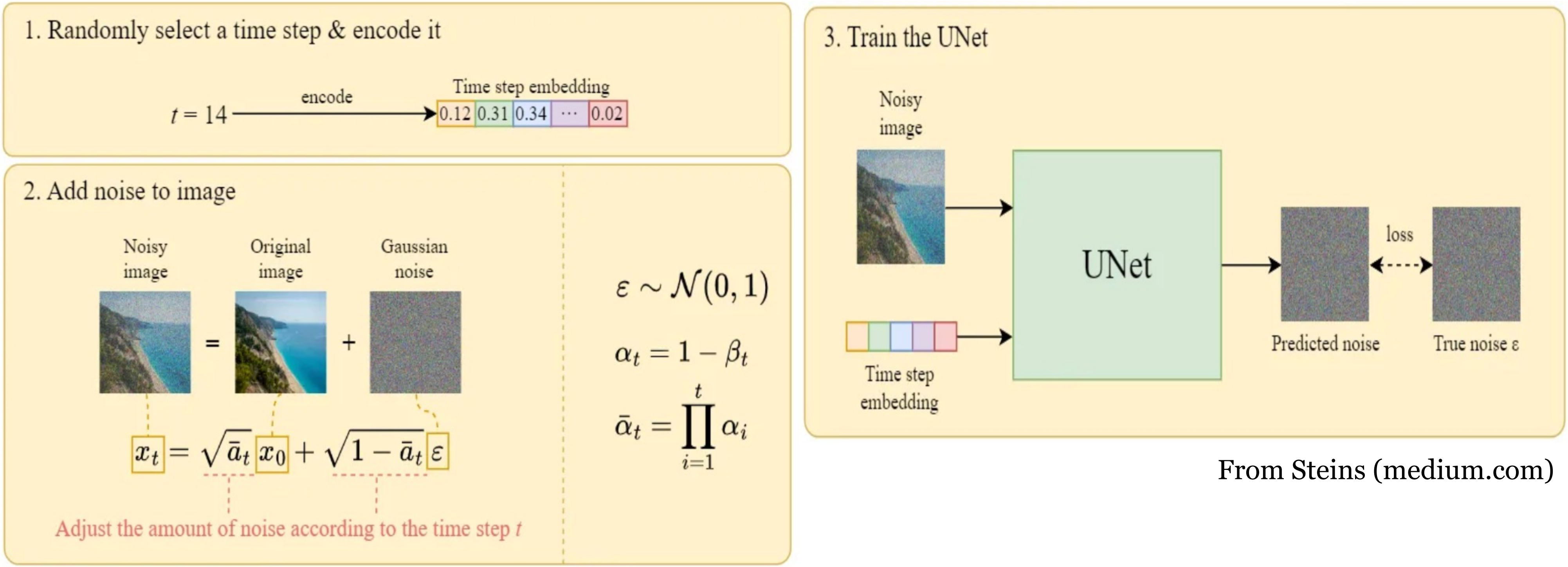
Repeat

- $\boldsymbol{x}_0 \sim q(\boldsymbol{x_0})$
- $t \sim uniform(\{1, \ldots, T\})$
- $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I})$
- $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
- $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} \left|\left|\epsilon - \epsilon_\theta(\boldsymbol{x}_t, t)\right|\right|_2^2$

Until convergence

Training Algorithm

For each training step:

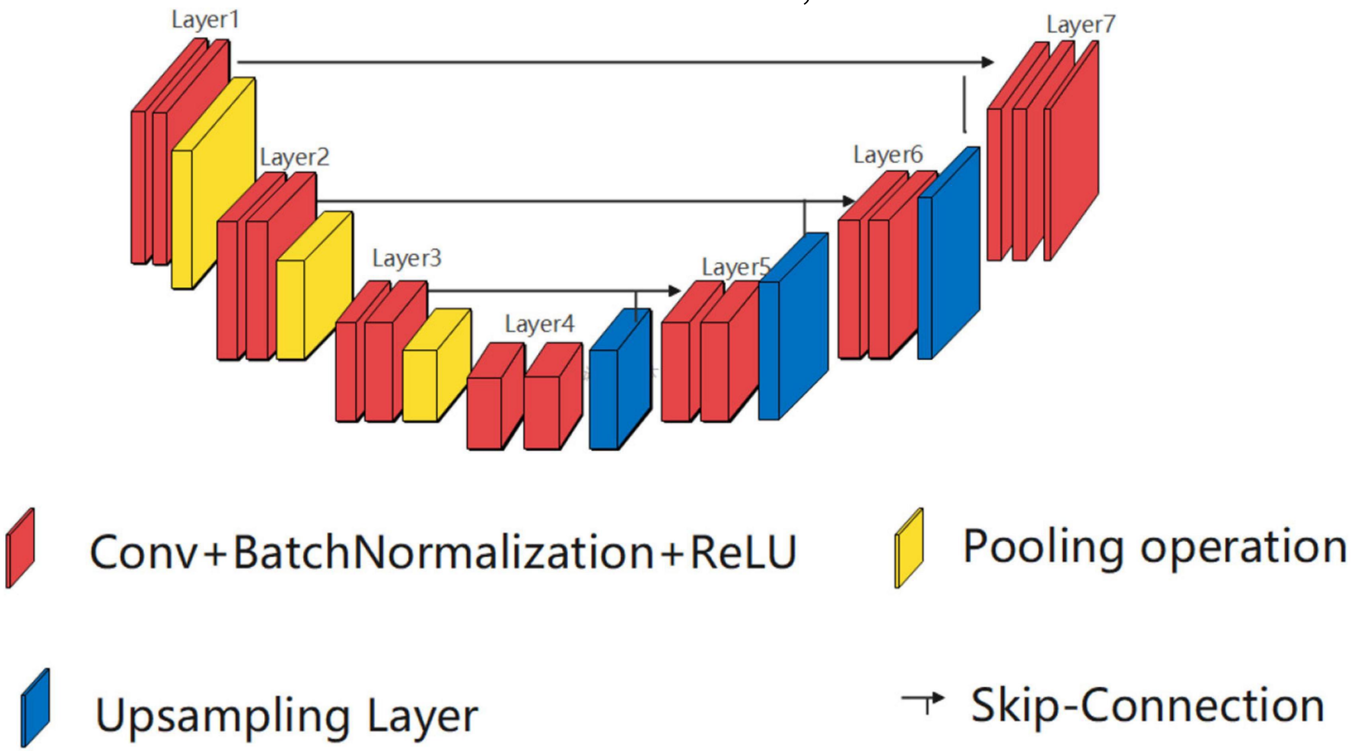


From Steins (medium.com)

UNet

Special type of fully convolutional neural network

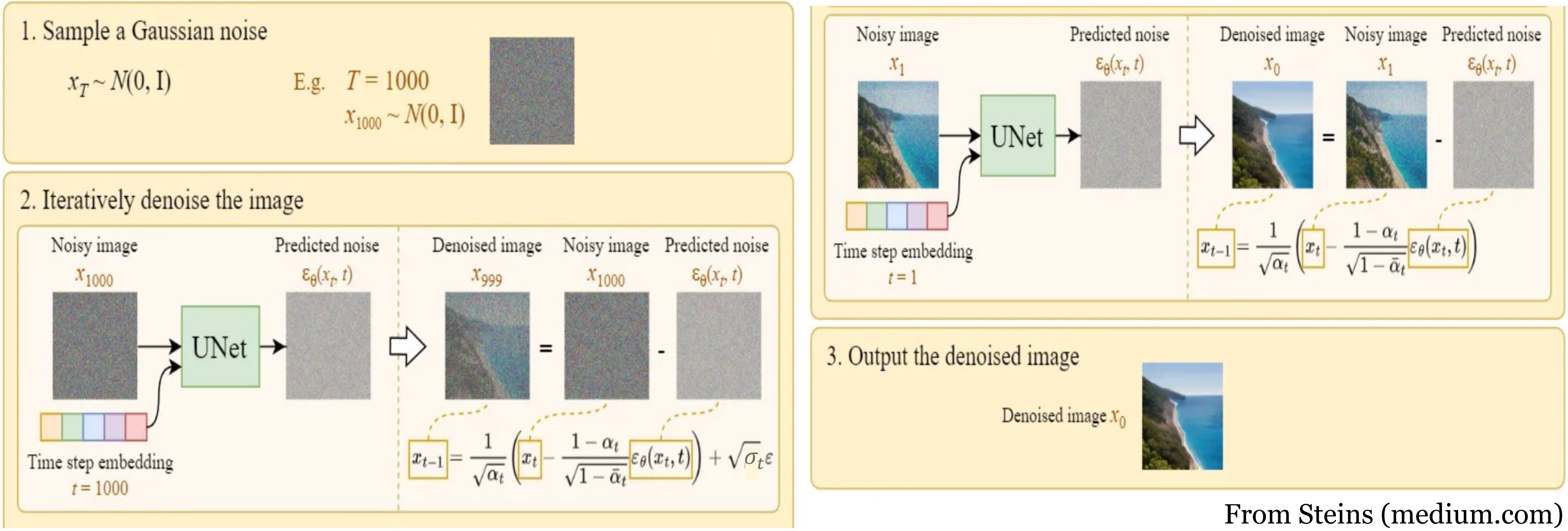
From Cai, Wu and Chen 2022



Data Generation Algorithm

- $\boldsymbol{x}_T \sim N(\boldsymbol{x}|\boldsymbol{0}, \boldsymbol{I})$
- For $t = T, \dots, 1$ do
  - $\boldsymbol{\epsilon} \sim N(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{\epsilon} = \boldsymbol{0}$
  - $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t)\right) + \sqrt{\sigma_t}\boldsymbol{\epsilon}$
- Return $\boldsymbol{x}_0$

Data Generation Algorithm



From Steins (medium.com)

. . .

Results

Ho, Jain and Abbeel (2020)

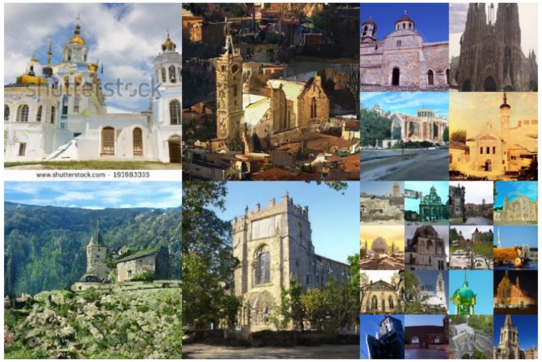


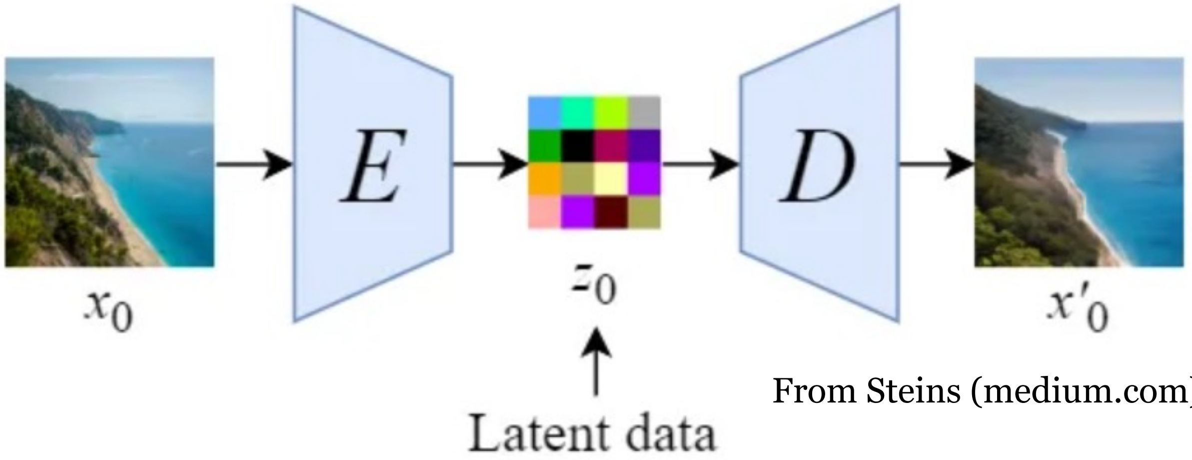
Figure 3: LSUN Church samples. FID=7.89

Figure 4: LSUN Bedroom samples. FID=4.90

Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

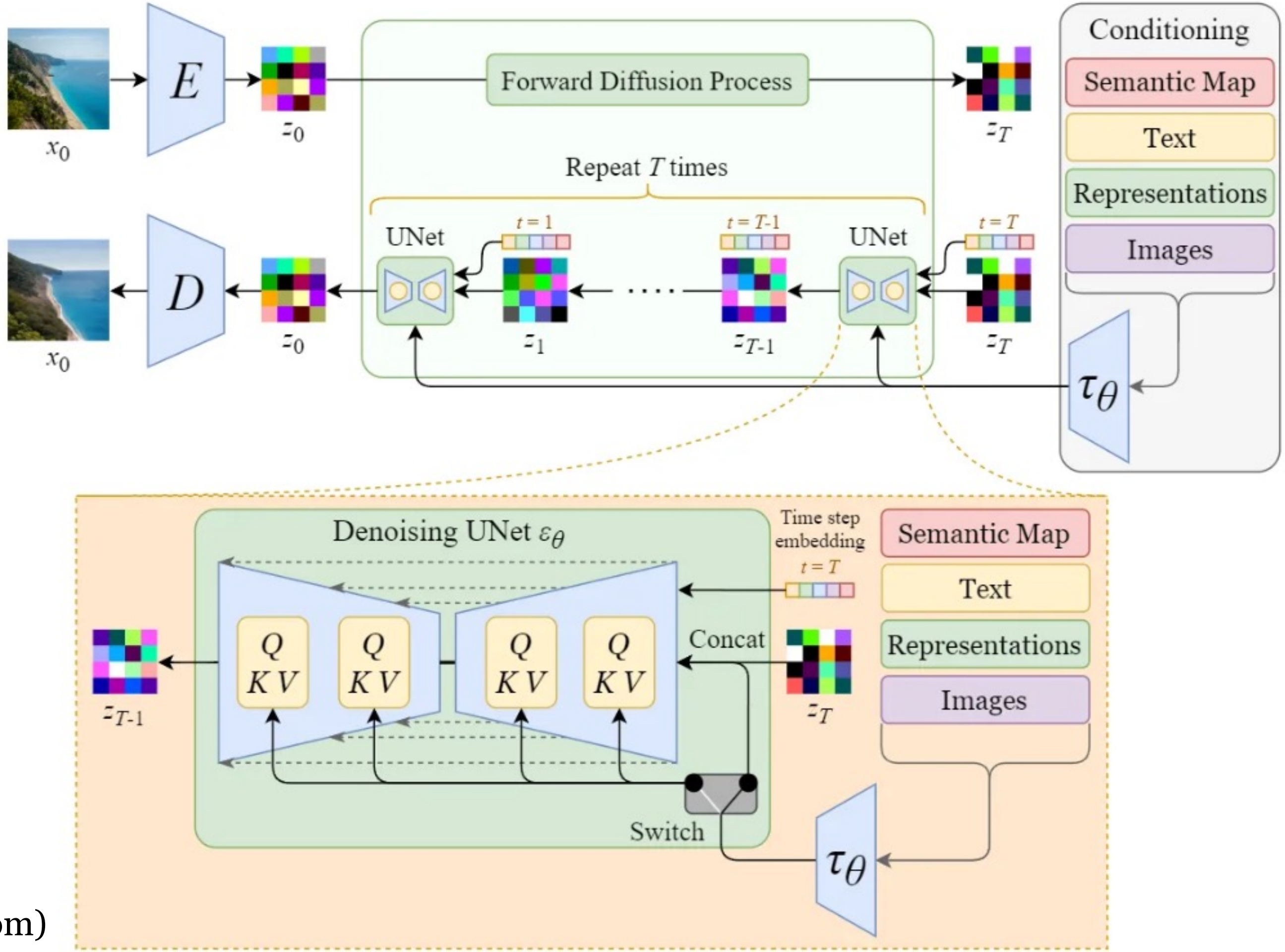
Latent Diffusion Model (a.k.a. Stable Diffusion)

- Rombach, Blattman et al., 2022
- **Speed up**: performing the diffusion in a low dimensional latent space
- **Conditional generation:** condition denoising on text, images, etc.



From Steins (medium.com)

Full Architecture



From Steins (medium.com)

Results

- Rombach, Blattman et al., 2022

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!" '*

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 $s = 10.0$.