Virtual Humans – Winter 24/25

Lecture 13_1 – Diffusion Models Theory

Prof. Dr.-Ing. Gerard Pons-Moll

University of Tübingen / MPI-Informatics

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

In this lecture…

- Recap of deep generative models
- Introduction of Diffusion Models
- Applications of Diffusion Models

Goal: **Generate** Virtual Humans

Generate Appearance

Generate Motion

So far we have seen…

- We can capture human appearance (NeRF, 3DGS)
- We can capture human motion (MoCap, Registration)
- We can capture human-object interaction (Behave, PHOSA, CHORE)

- **Can we also generate the appearance and motion of ”human”?**
- **Why is "synthesis" of Virtual Human useful?**

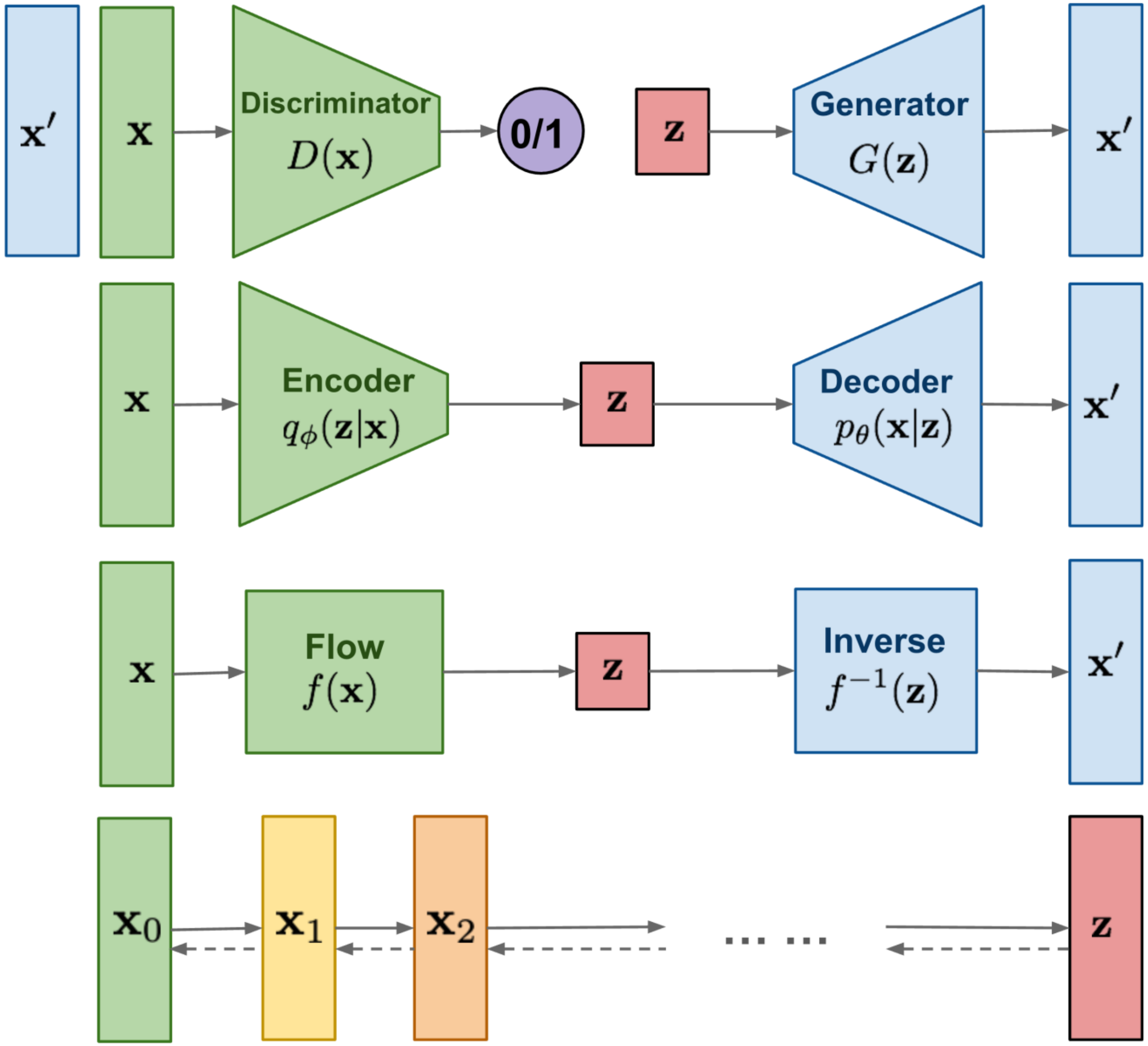
Generative Models

GAN: Adversarial training

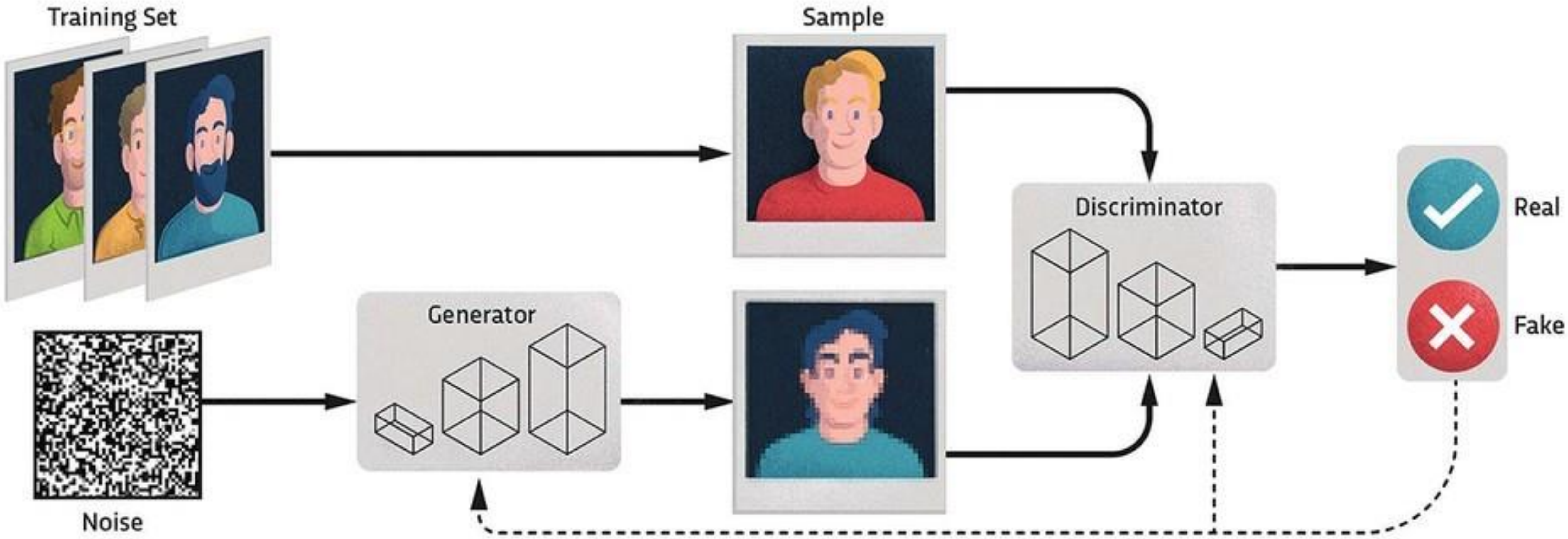
VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse

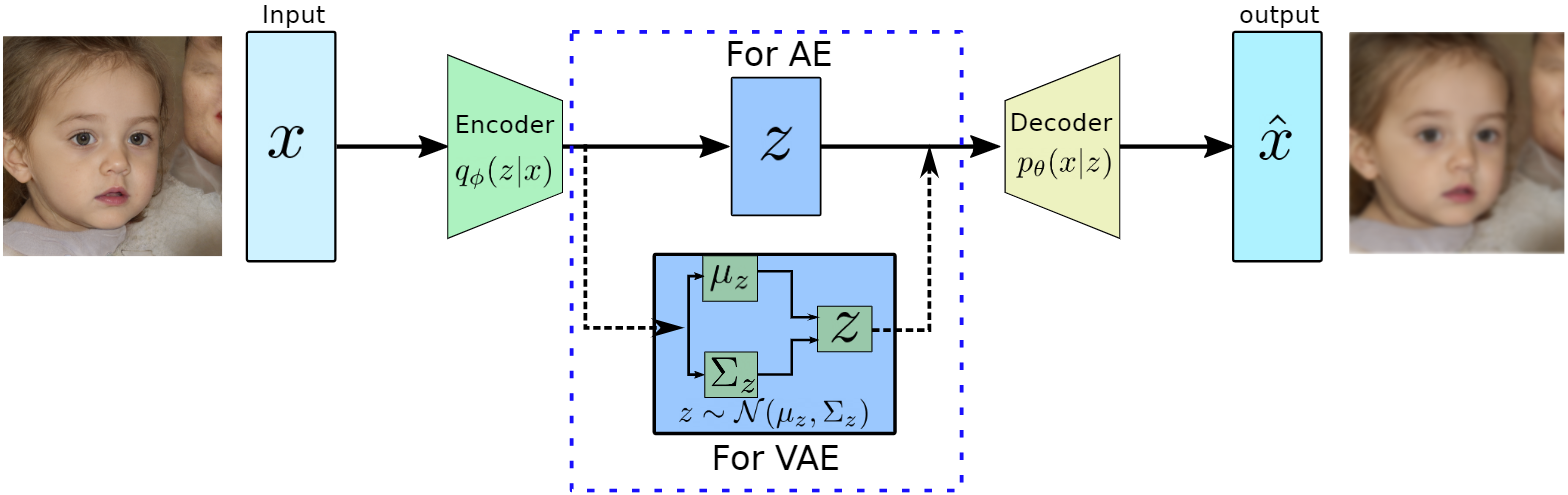


GAN



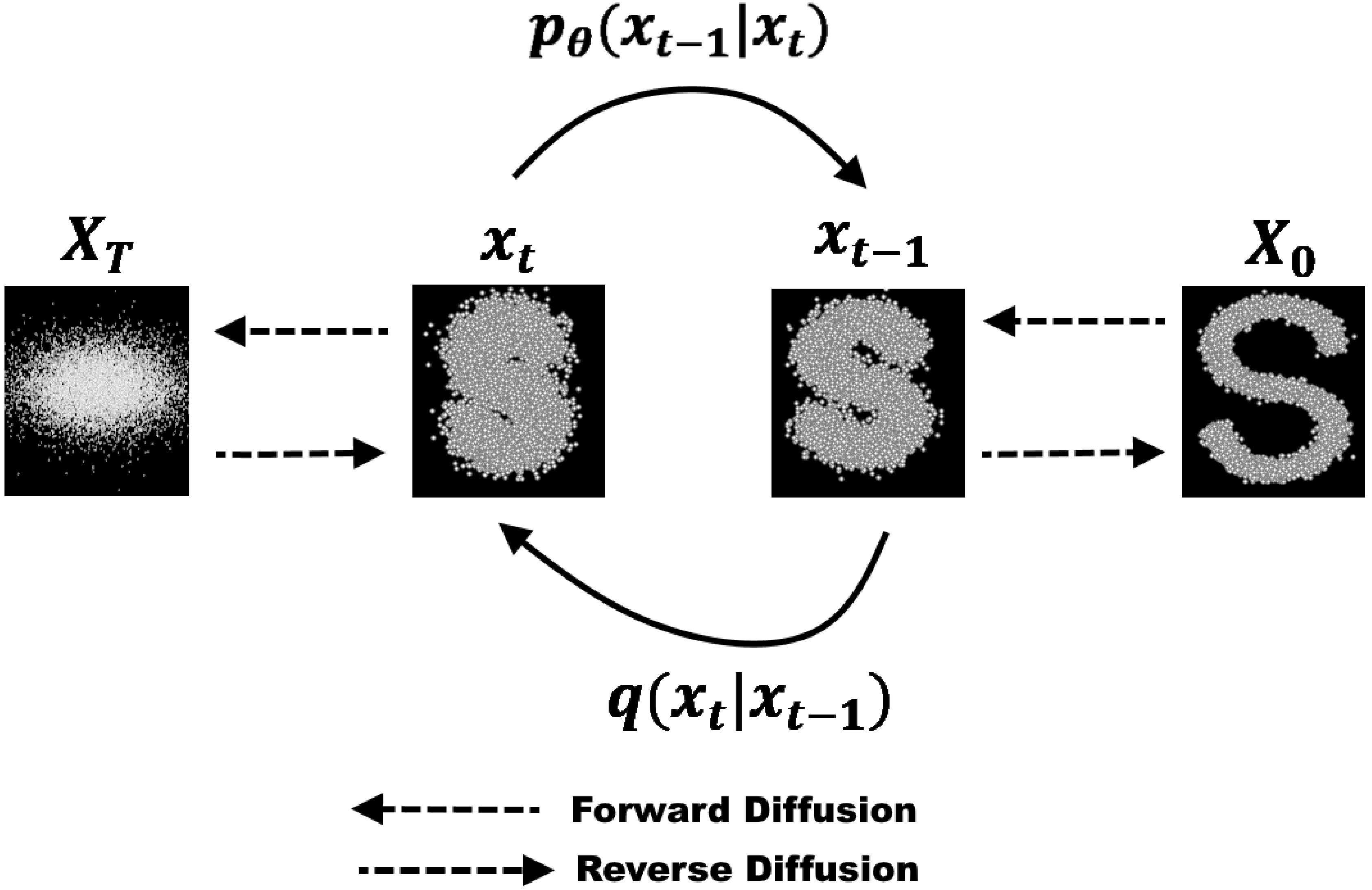
Training Set
Sample
Discriminator
Real
Fake
Generator
Noise

VAE

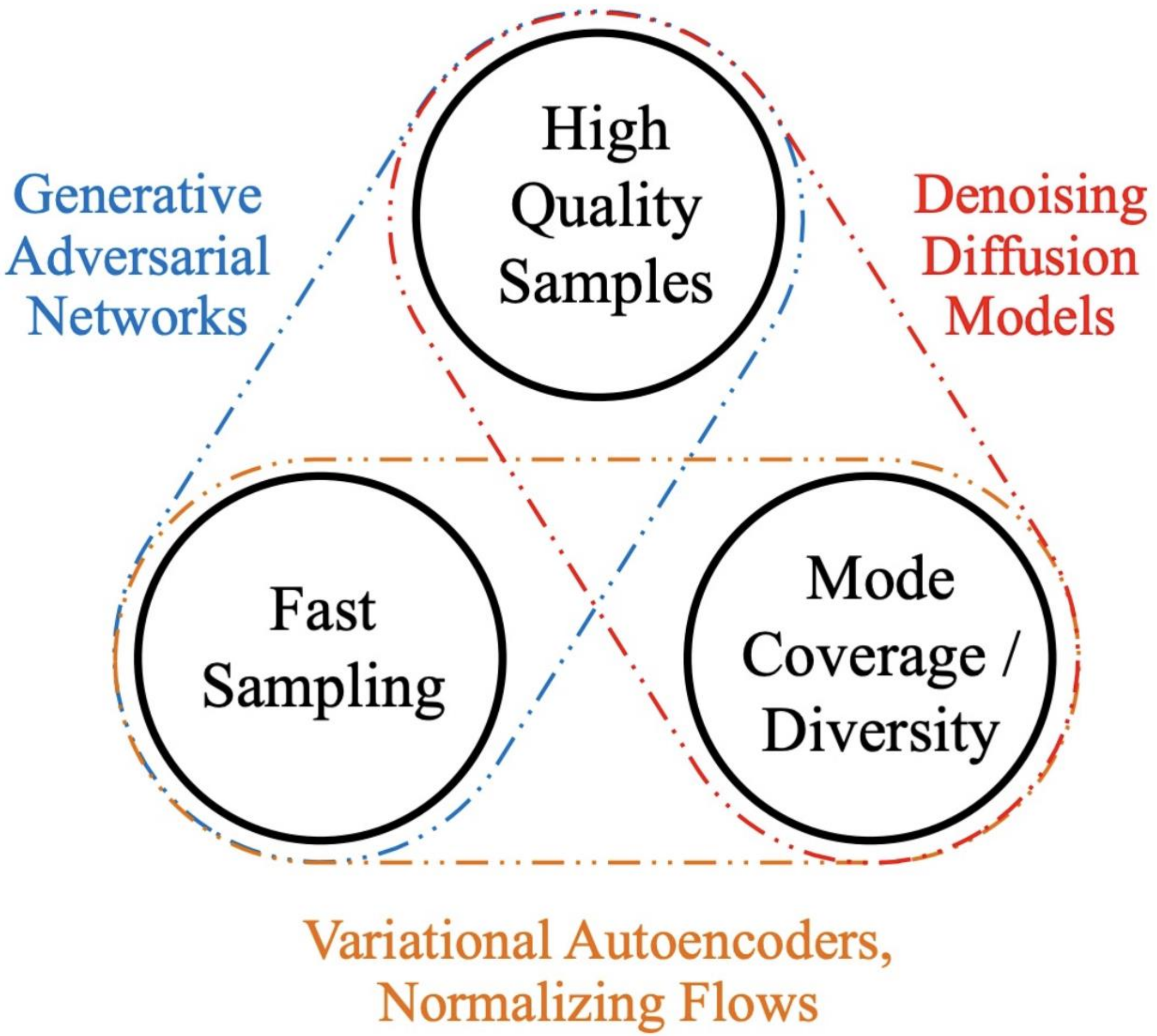


Input
x
Encoder
$q_\phi(z|x)$
For AE
z
μ_z
Σ_z
z
$z \sim \mathcal{N}(\mu_z, \Sigma_z)$
For VAE
Decoder
$p_\theta(x|z)$
output
$\hat{x}$

Diffusion Models



Trilemma: Quality, Diversity Speed



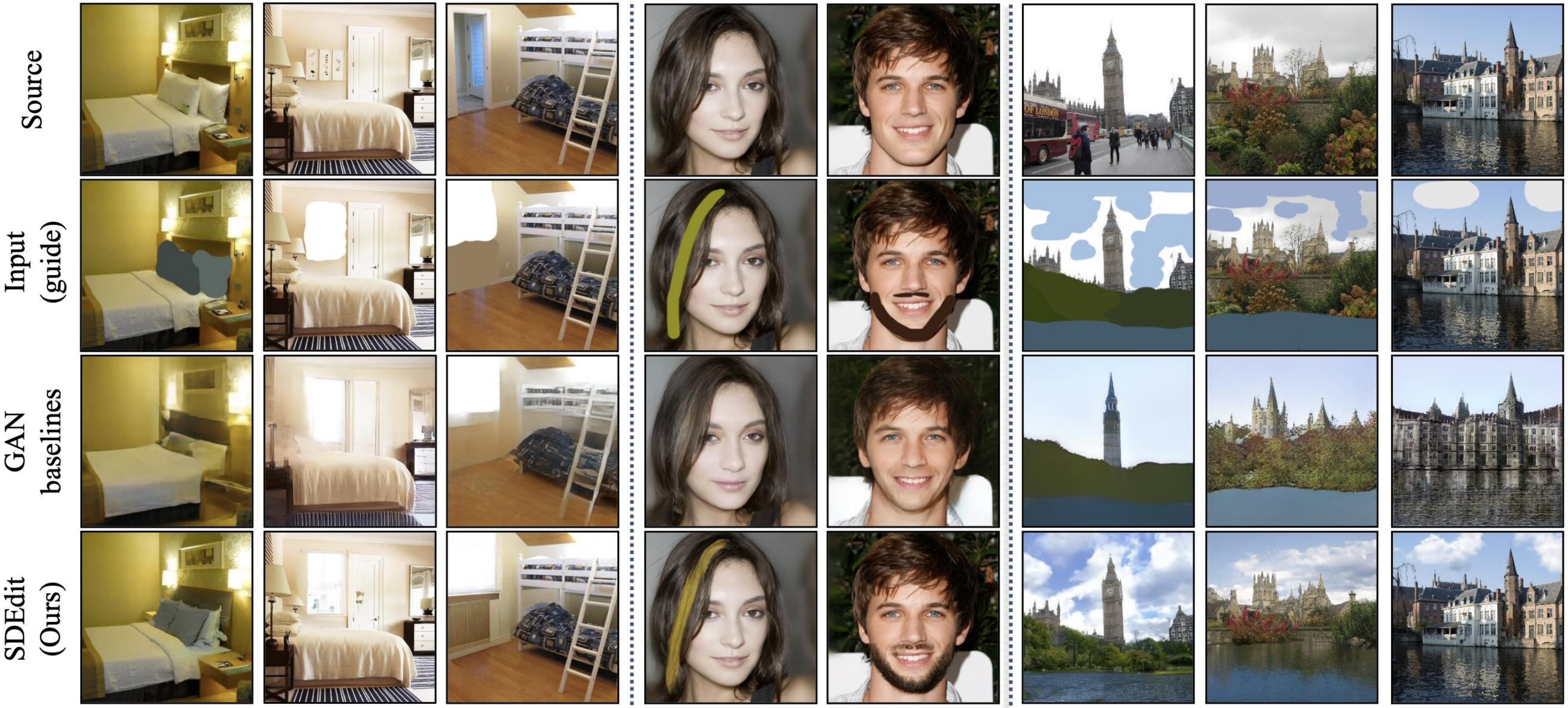
Diffusion Model for Image Generation

- Diffusion model is SOTA on image generation

- Stable Diffusion
- Mid-Journey
- Flux
- …

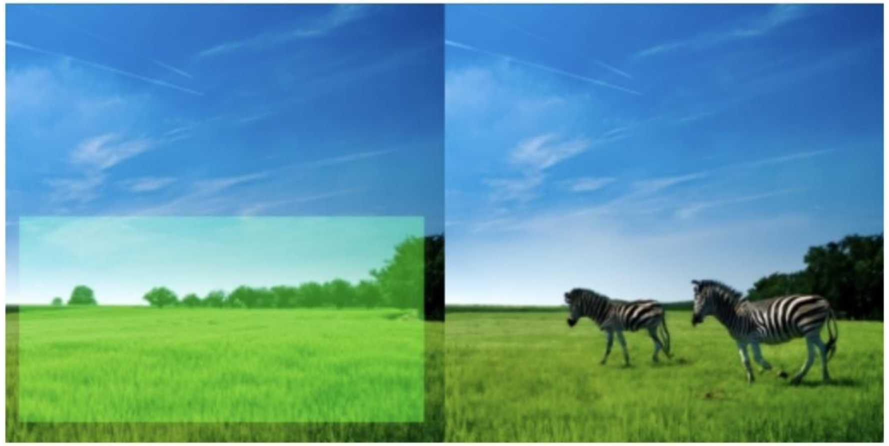
Diffusion Model for Image Generation

- Diffusion model is useful for image editing



Diffusion Model for Image Generation

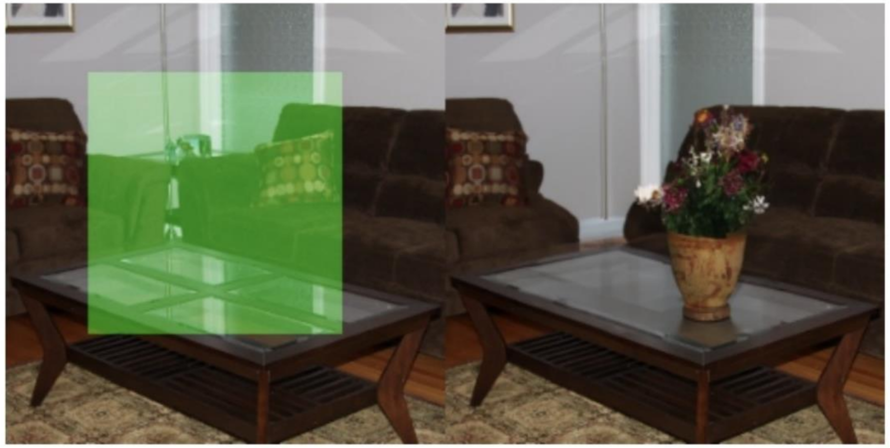
- Diffusion model is useful for image editing



"zebras roaming in the field"

"a girl hugging a corgi on a pedestal"

"a man with red hair"



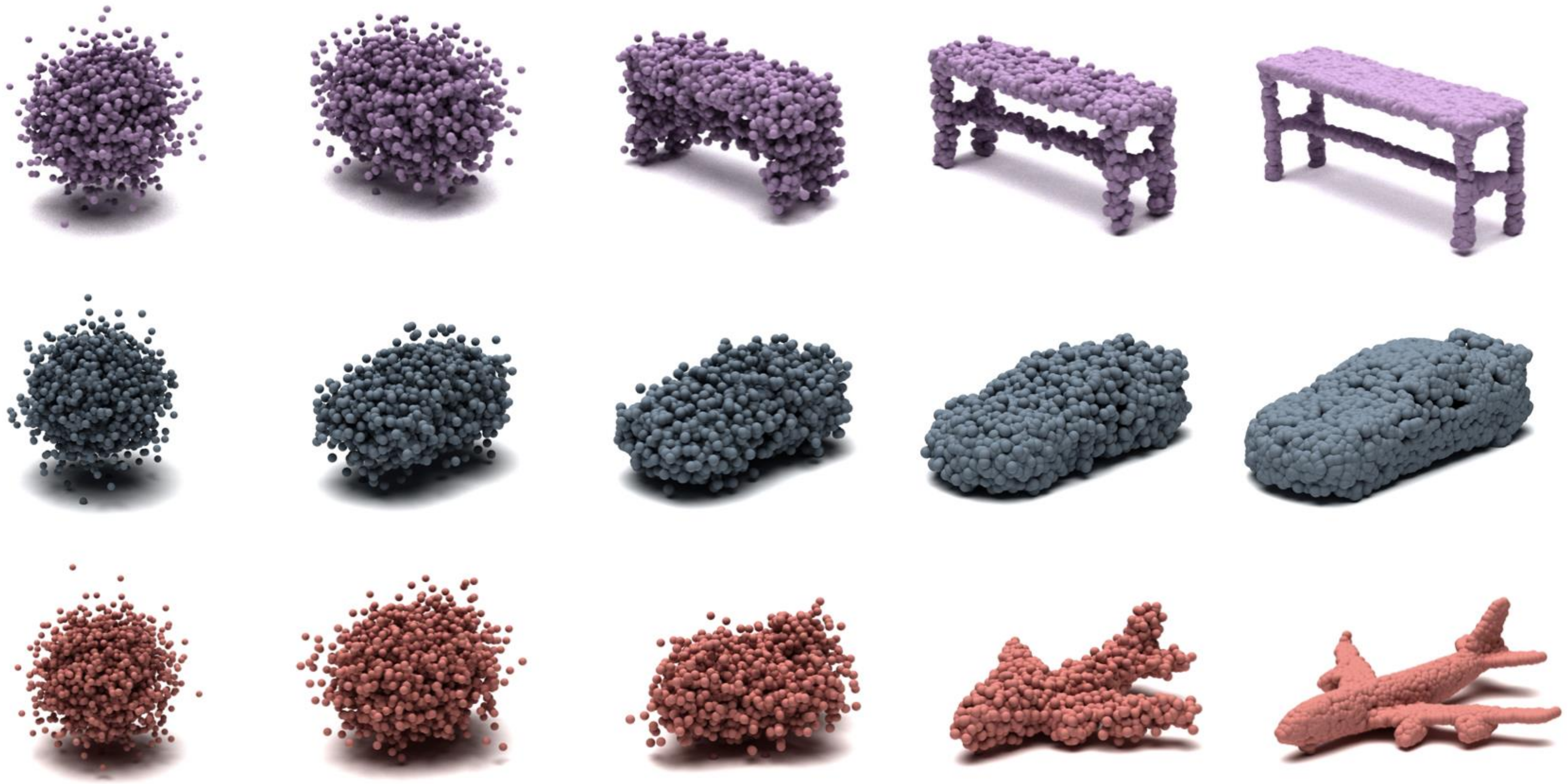
"a vase of flowers"

Diffusion Model for Image Generation

- Diffusion model is applicable for other non-visual domains
  - Generate motion from text

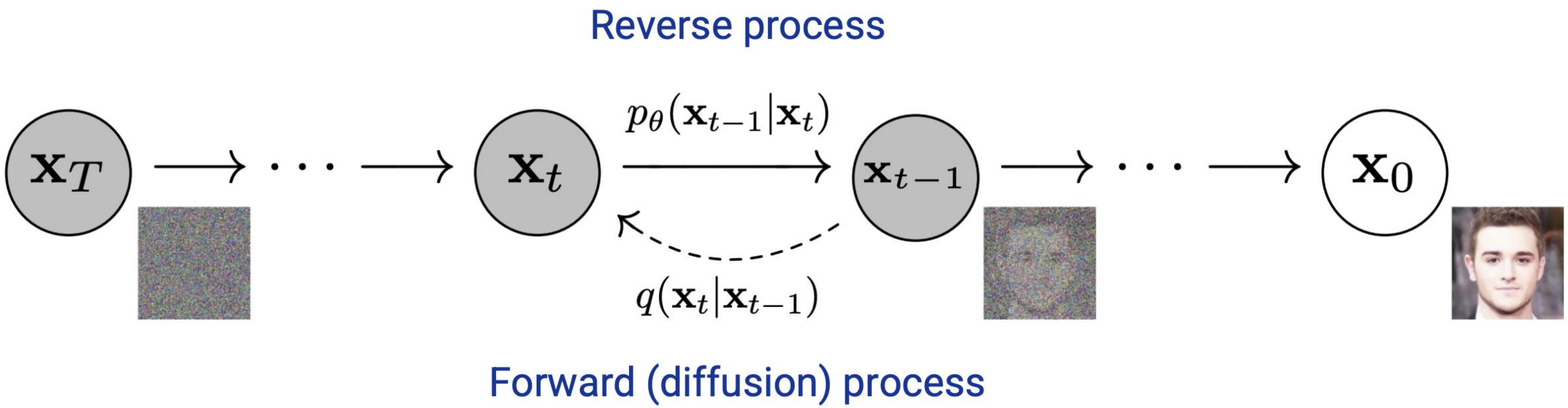
Diffusion Model for Image Generation

- Diffusion model is applicable for other non-visual domains
  - Generate 3D point cloud



Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Forward step**: (Iteratively) Add noise to the original sample
    - → The sample $x!$ converges to the complete noise x_T (e.g., $\sim \mathcal{N}(0, I)$)
  - **Reverse step**: Recover the original sample from the noise
    - → Note that it is the "generation" procedure



Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Forward step**: (Iteratively) Add noise to the original sample
    - → Technically, it is a product of conditional noise distributions $q(\mathbf{x}_t|\mathbf{x}_{t-1})$
    - Usually, the parameters β_t are fixed (one can jointly learn, but not beneficial)
    - • Noise annealing (i.e., reducing noise scale $\beta_t < \beta_{t-1}$) is crucial to the performance

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Forward step**: (Iteratively) Add noise to the original sample
    - → Technically, it is a product of conditional noise distributions $q(\mathbf{x}_t|\mathbf{x}_{t-1})$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

  - **Reverse step**: Recover the original sample from the noise
    - → It is also a product of conditional (de)noise distributions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
    - • Use the learned parameters: denoiser $\boldsymbol{\mu}_\theta$ (main part) and randomness $\boldsymbol{\Sigma}_\theta$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Forward step**: (Iteratively) Add noise to the original sample
  - **Reverse step**: Recover the original sample from the noise

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$

  - **Training**: Minimize variational lower bound of the model $p_\theta(\mathbf{x}_0)$

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$$

Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Training**: Minimize variational lower bound of the model $p_\theta(\mathbf{x}_0)$

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$$

    - → It can be decomposed to the step-wise losses (for each step t)

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

Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Training**: Minimize variational lower bound of the model $p_\theta(\mathbf{x}_0)$
    - → It can be decomposed to the step-wise losses (for each step t)

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

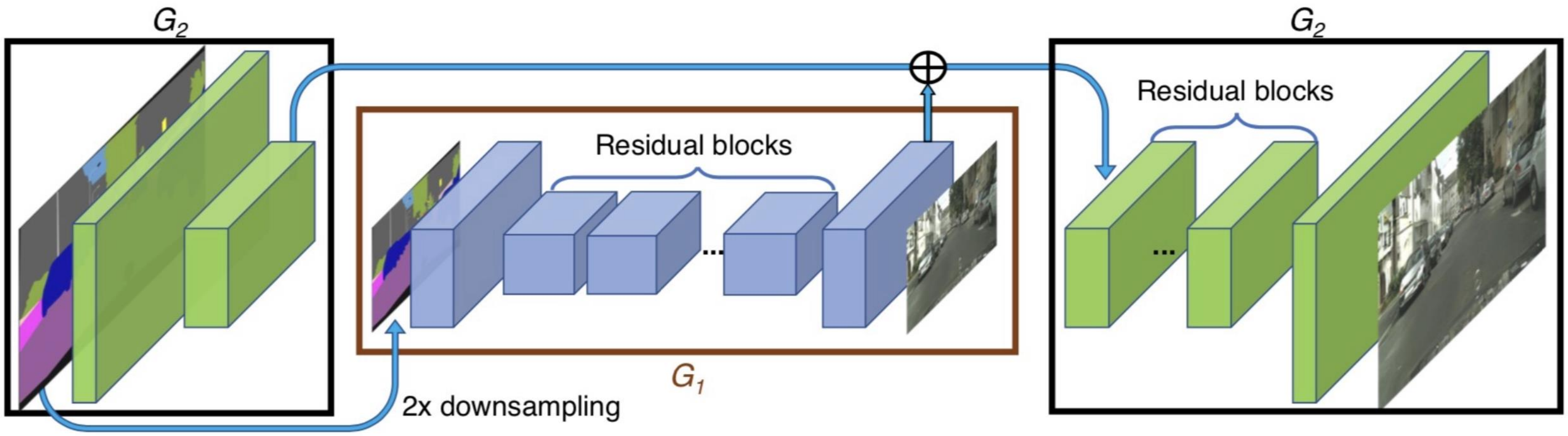
  - Here, the true reverse step $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be computed as a closed form of β_t
  - Note that we only define the true forward step $q(\mathbf{x}_t|\mathbf{x}_{t-1})$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_{t-1};\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0),\tilde{\beta}_t^3\mathbf{I})$$

$$\text{where}\quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0):=\tilde{\beta}_t^1\mathbf{x}_0+\tilde{\beta}_t^2\mathbf{x}_t$$

Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Network:** Use the image-to-image translation (e.g., U-Net) architectures
    - Recall that input is $\mathbf{x}$t and output is $\mathbf{x}$t-1, both are images
    - • It is expensive since both input and output are high-dimensional
    - • Note that the denoiser $\mu_\theta(\mathbf{x}_t$, t) conditioned by step t

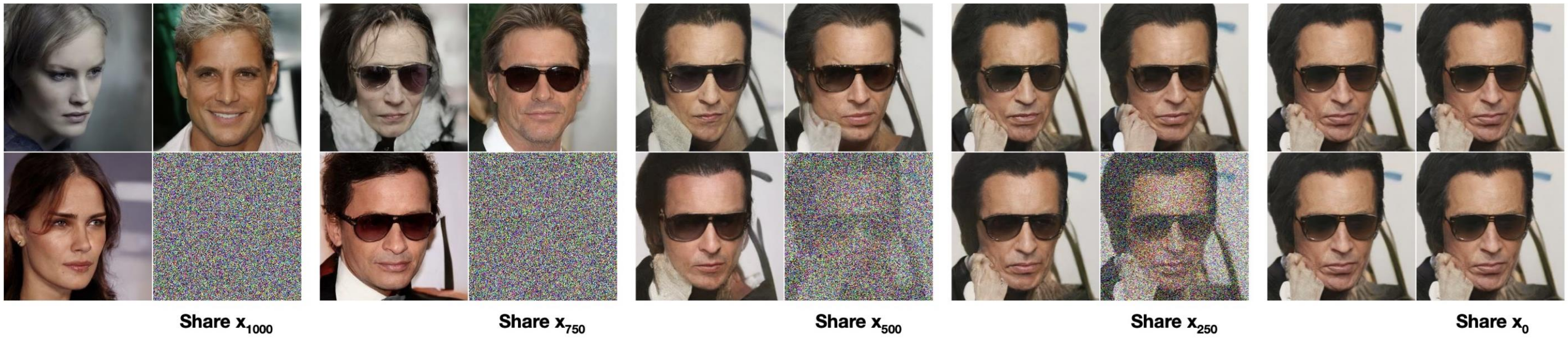


Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Sampling**: Draw a random noise $\boldsymbol{x}_T$ then apply the reverse step $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
    - It often requires the 1000 reverse steps (very slow)

Diffusion Probabilistic Models

- Diffusion model aims to learn the **reverse** of noise generation procedure
  - **Sampling**: Draw a random noise $\boldsymbol{x}_{\mathrm{T}}$ then apply the reverse step $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
    - Early and late steps change the high- and low-level attributes, respectively



Denoising Diffusion Probabilistic Models

- DDPM reparametrizes the reverse distributions of diffusion models
  - **Key idea**: The original reverse step fully creates the denoiser $\mu_\theta(\mathbf{x}_t, t)$
    - However, $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ share most information, and thus it is redundant
    - → Instead, create the **residual** $\epsilon_\theta(\mathbf{x}_t, t)$ and add to the original $\mathbf{x}_t$
  - Formally, DDPM reparametrizes the learned reverse distribution as

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$$

  - and the step-wise objective L_{t-1} can be reformulated as

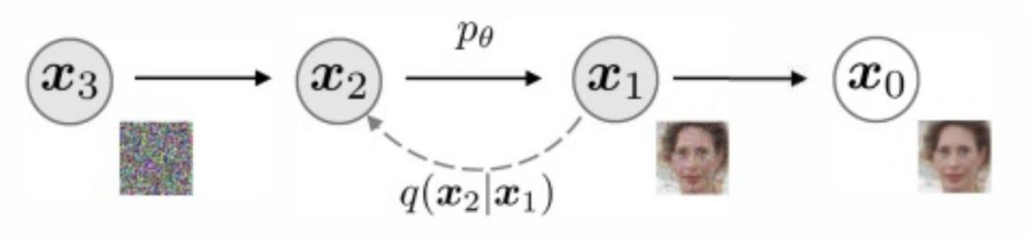
$$\mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

Denoising Diffusion Implicit Models

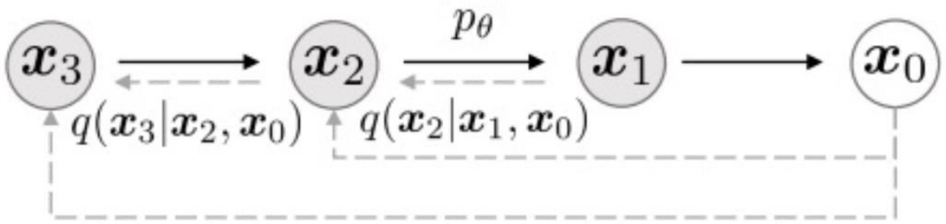
- DDIM roughly sketches the final sample, then refine it with the reverse process
  - **Motivation**:
    - Diffusion model is slow due to the iterative procedure
    - GAN/VAE creates the sample by one-shot forward operation
    - ⇒ Can we combine the advantages for **fast sampling** of diffusion models?
  - **Technical spoiler**:
    - Instead of naïvely applying diffusion model upon GAN/VAE,
    - DDIM proposes a principled approach of rough sketch + refinement

Denoising Diffusion Implicit Models

- DDIM roughly sketches the final sample, then refine it with the reverse process
  - **Key Idea**:
    - Given $\mathbf{x}_t$, generate the rough sketch $\mathbf{x}_0$ and refine $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
    - Unlike original diffusion model, it is not a Markovian structure



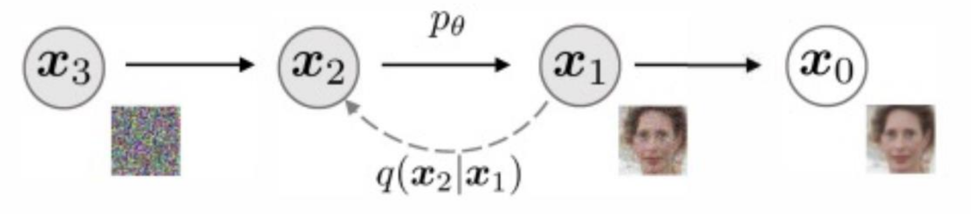
Original Diffusion

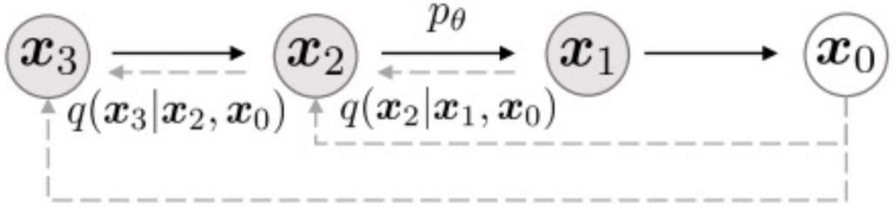


Non-Markovian

Denoising Diffusion Implicit Models

- DDIM roughly sketches the final sample, then refine it with the reverse process
  - **Key Idea**: Given $\mathbf{x}_t$, generate the rough sketch $\mathbf{x}_0$ and refine $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$





  - **Formulation:** Define the forward distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ as

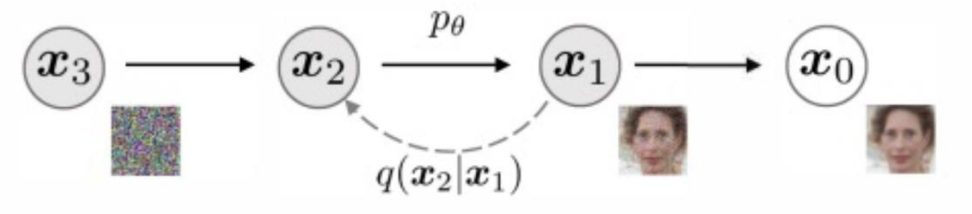
$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \boldsymbol{I}\right)$$

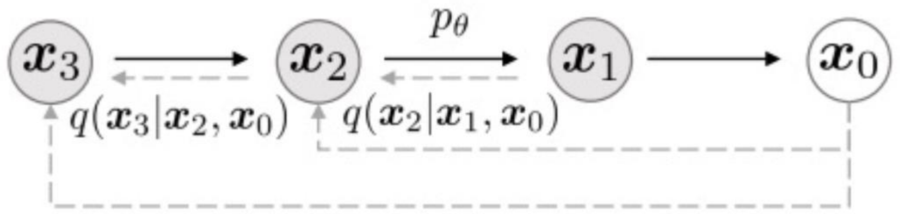
then, the forward process is derived from Bayes' rule

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \frac{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}$$

Denoising Diffusion Implicit Models

- DDIM roughly sketches the final sample, then refine it with the reverse process
  - **Key Idea**: Given $\mathbf{x}_t$, generate the rough sketch $\mathbf{x}_0$ and refine $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$





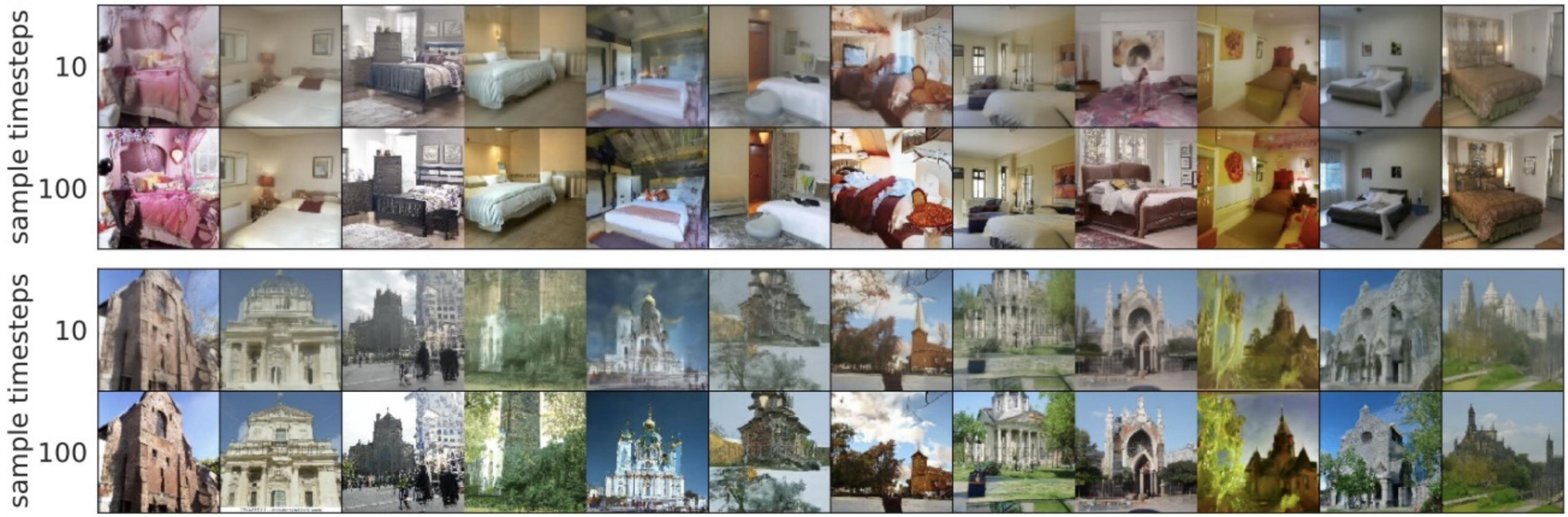
  - **Formulation:** Forward process is $q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \dfrac{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}$

and reverse process is

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}}\right)}_{\text{" predicted } \boldsymbol{x}_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{"direction pointing to } \boldsymbol{x}_t\text{"}} + \underbrace{\sigma_t\epsilon_t}_{\text{random noise}}$$

Denoising Diffusion Implicit Models

- DDIM significantly reduces the sampling steps of diffusion model
  - Creates the outline of the sample after only 100 steps (DDPM needs thousands)



Takeaways

- **New golden era** of generative models
- Competition of various approaches: GAN, VAE, flow, diffusion model
- **Diffusion model** seems to be a nice option for **high-quality generation**