Introduction to Diffusion Models

2022.01.03.

KAIST ALIN-LAB

Sangwoo Mo

Diffusion Model Boom!

- **Diffusion model is SOTA on image generation**
  - Beat BigGAN and StyleGAN on high-resolution images

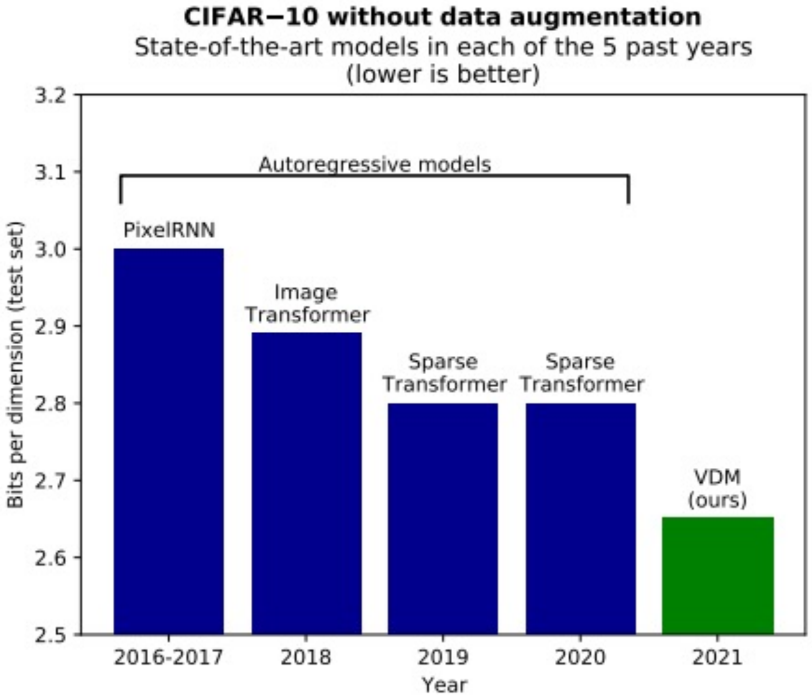
Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Model	FID	sFID	Prec	Rec
LSUN Bedrooms 256×256				
DCTransformer† [42]	6.40	6.66	0.44	**0.56**
DDPM [25]	4.89	9.07	0.60	0.45
IDDPM [43]	4.24	8.21	0.62	0.46
StyleGAN [27]	2.35	6.62	0.59	0.48
ADM (dropout)	**1.90**	**5.59**	**0.66**	0.51
ImageNet 512×512				
BigGAN-deep [5]	8.43	8.13	**0.88**	0.29
ADM	23.24	10.19	0.73	**0.60**
ADM-G (25 steps)	8.41	9.67	0.83	0.47
ADM-G	**7.72**	**6.57**	0.87	0.42

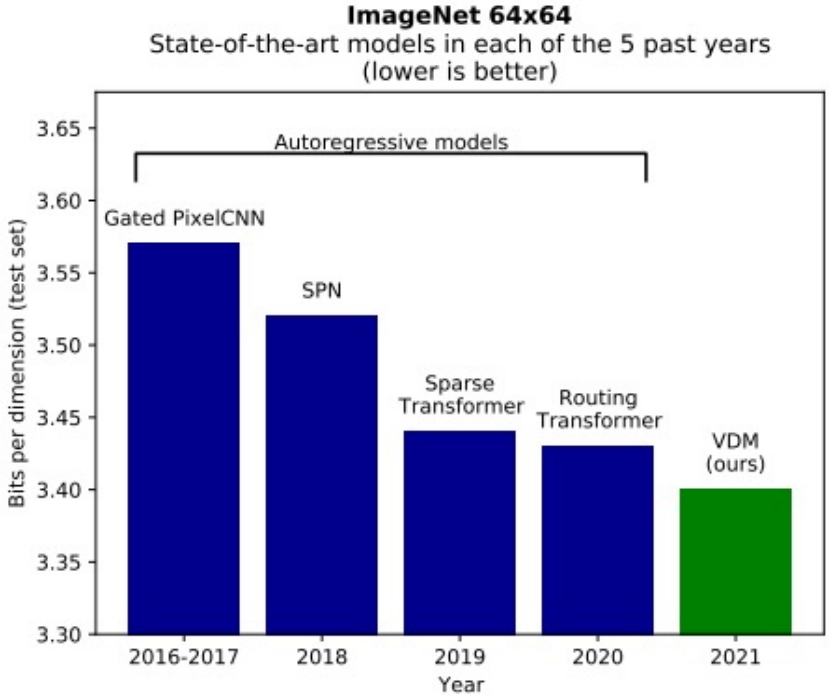
Dhariwal & Nichol. Diffusion Models Beat GANs on Image Synthesis. NeurIPS’21

Diffusion Model Boom!

- **Diffusion model is SOTA on density estimation**
  - Beat autoregressive models on likelihood score



(a) CIFAR-10 without data augmentation

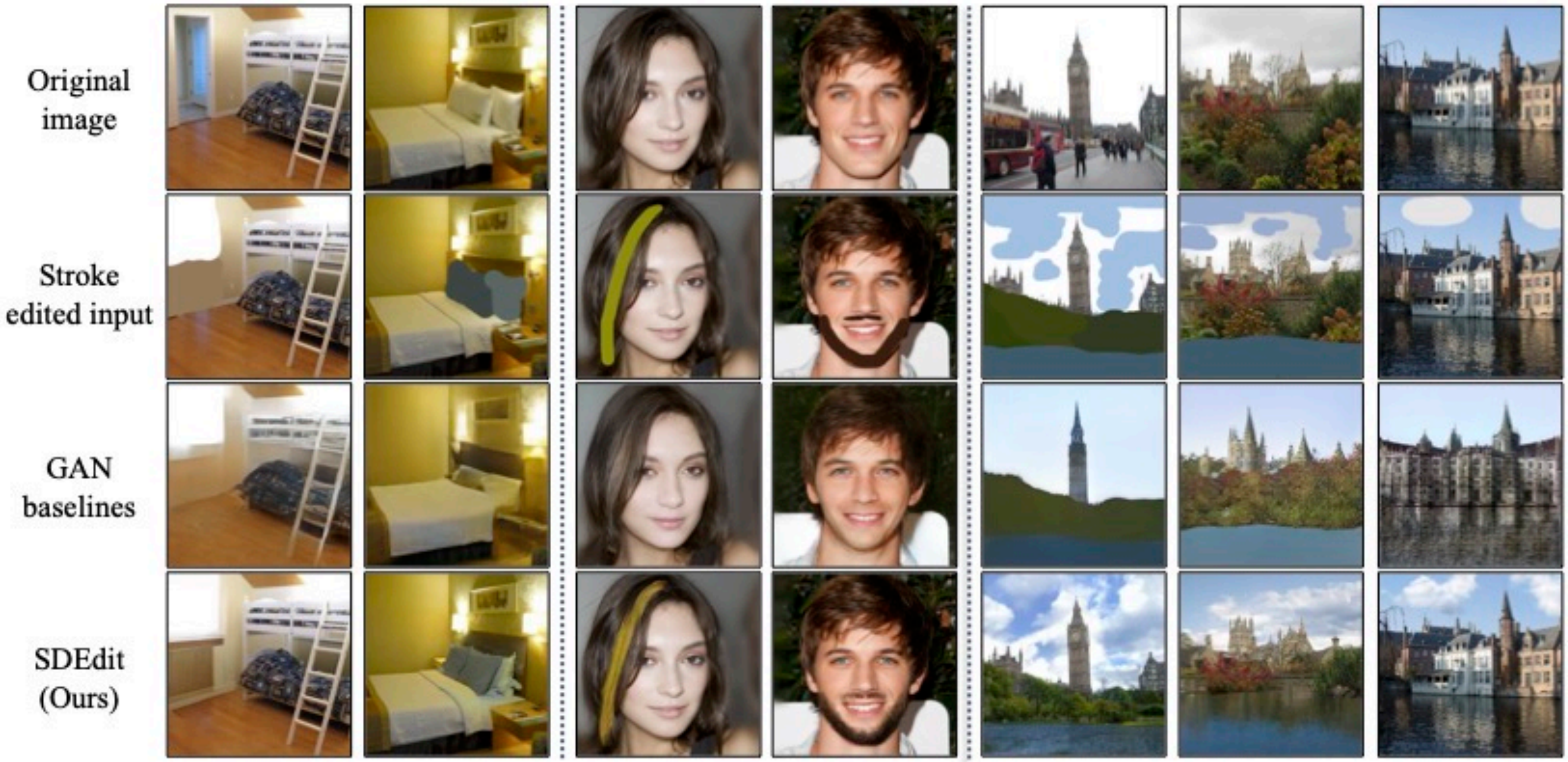


(b) ImageNet 64x64

Song et al. Maximum Likelihood Training of Score-Based Diffusion Models. NeurIPS’21
Kingma et al. Variational Diffusion Models. NeurIPS’21

Diffusion Model Boom!

- **Diffusion model is useful for image editing**
  - Editing = Rough scribble + diffusion (i.e., naturalization)
  - Scribbled images are unseen for GANs, but diffusion models still can *denoise* them



Meng et al. SDEdit: Image Synthesis and Editing with Stochastic Differential Equations. arXiv'21

Diffusion Model Boom!

- **Diffusion model is useful for image editing**
    - Also can be combined with vision-and-language model

“zebras roaming in the field”

“a girl hugging a corgi on a pedestal”

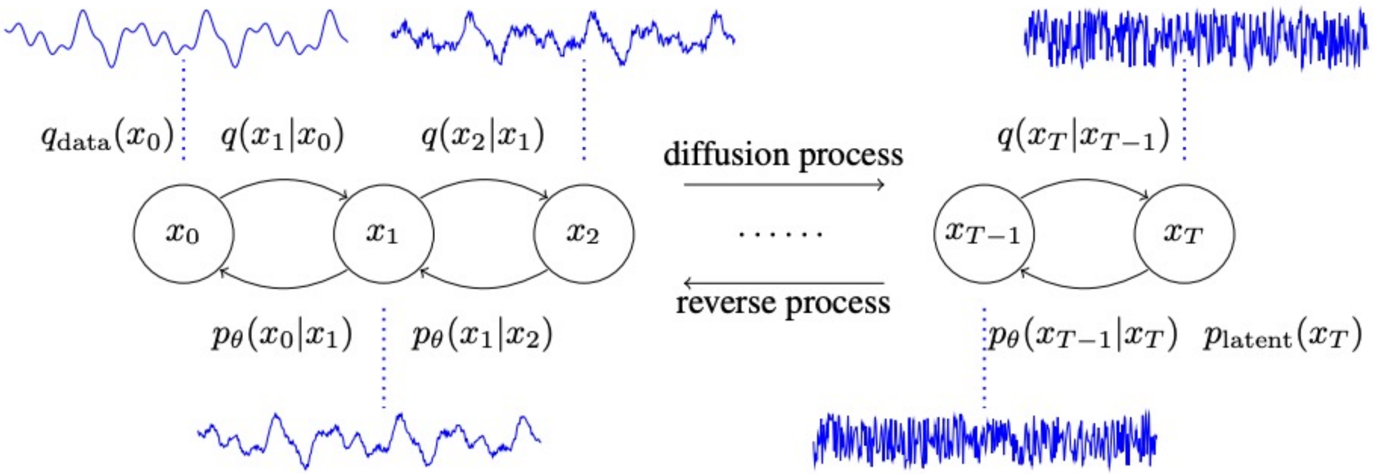
“a man with red hair”

“a vase of flowers”

Nichol et al. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. arXiv’21

Diffusion Model Boom!

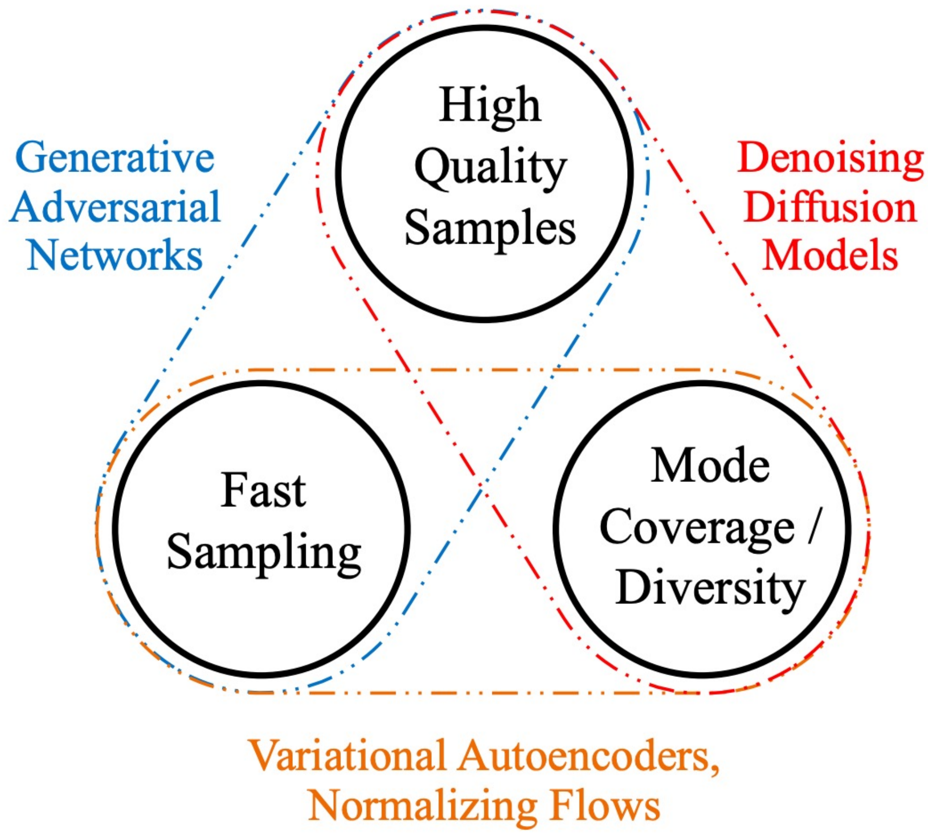
- **Diffusion model is also effective for non-visual domains**
  - Continuous domains like **speech**, and even for discrete domains like **text**



Kong et al. DiffWave: A Versatile Diffusion Model for Audio Synthesis. ICLR'21
Austin et al. Structured Denoising Diffusion Models in Discrete State-Spaces. NeurIPS'21

Diffusion Model is All We Need?

- **Trilemma of generative models: Quality vs. Diversity vs. Speed**
  - Diffusion model produces diverse and high-quality samples, but generations is slow



Xiao et al. Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. arXiv'21

Outline

- **Today's content**
  - Diffusion Probabilistic Model – ICML'15
  - Denoising Diffusion Probabilistic Model (DDPM) – NeurIPS'20
    - Improve quality & diversity of diffusion model
  - Denoising Diffusion Implicit Model (DDIM) – ICLR'21
    - Improve generation speed of diffusion model

- **Not covering**
  - Relation of diffusion model and score matching
  - Extension to stochastic differential equation

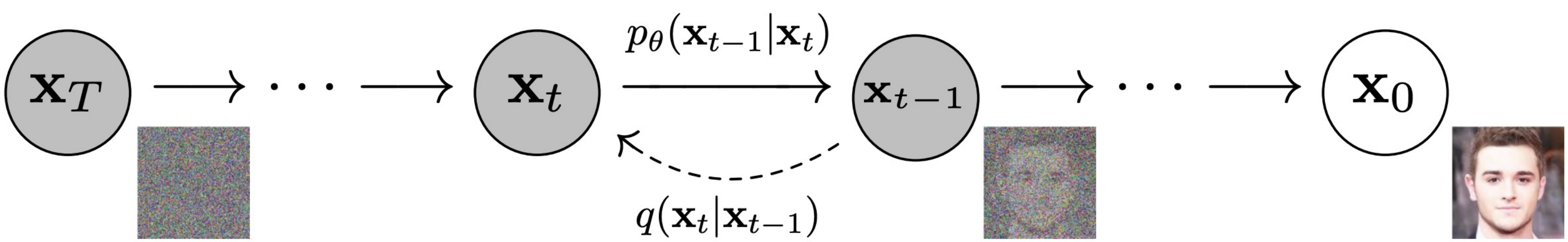
  → See **Score SDE** (ICLR'21)
  - There are lots of new interesting works (see NeurIPS'21, ICLR'22)

Score SDE: Song et al. Score-Based Generative Modeling through Stochastic Differential Equations. ICLR'21

Diffusion Probabilistic Model

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

    → The sample x_0 converges to the complete noise x_T (e.g., $\sim \mathcal{N}(0, I)$)



Forward (diffusion) process

Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

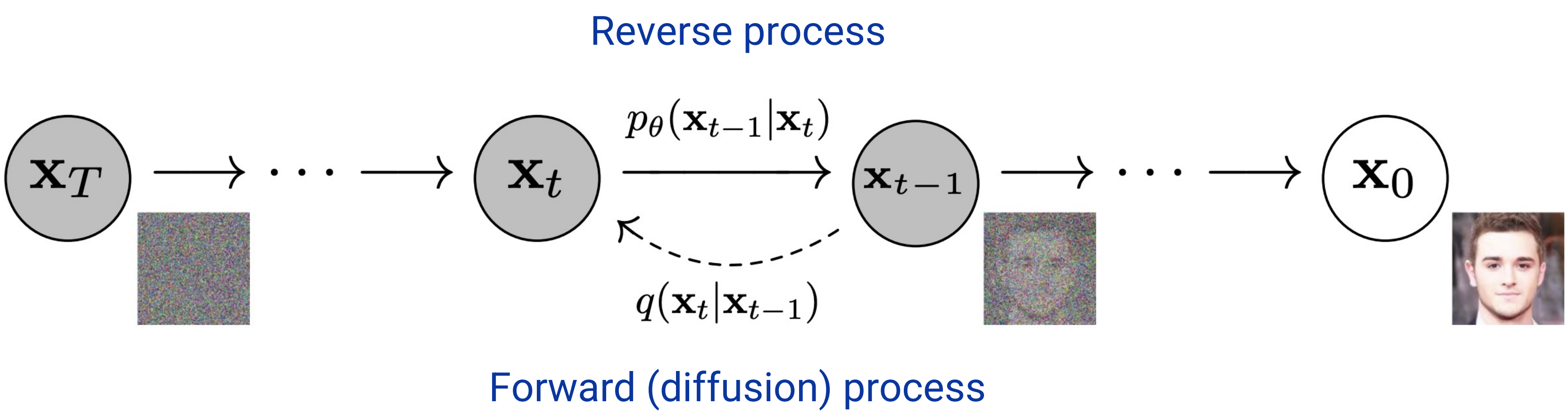
Diffusion Probabilistic Model

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

    → The sample x_0 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



Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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

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

Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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 $\boldsymbol{\Sigma}_\theta$

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

Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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 $p_\theta(\mathbf{x}_0)$

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

Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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 $p_\theta(\mathbf{x}_0)$

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

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

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

Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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 $q(\mathbf{x}_t|\mathbf{x}_{t-1})$

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

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

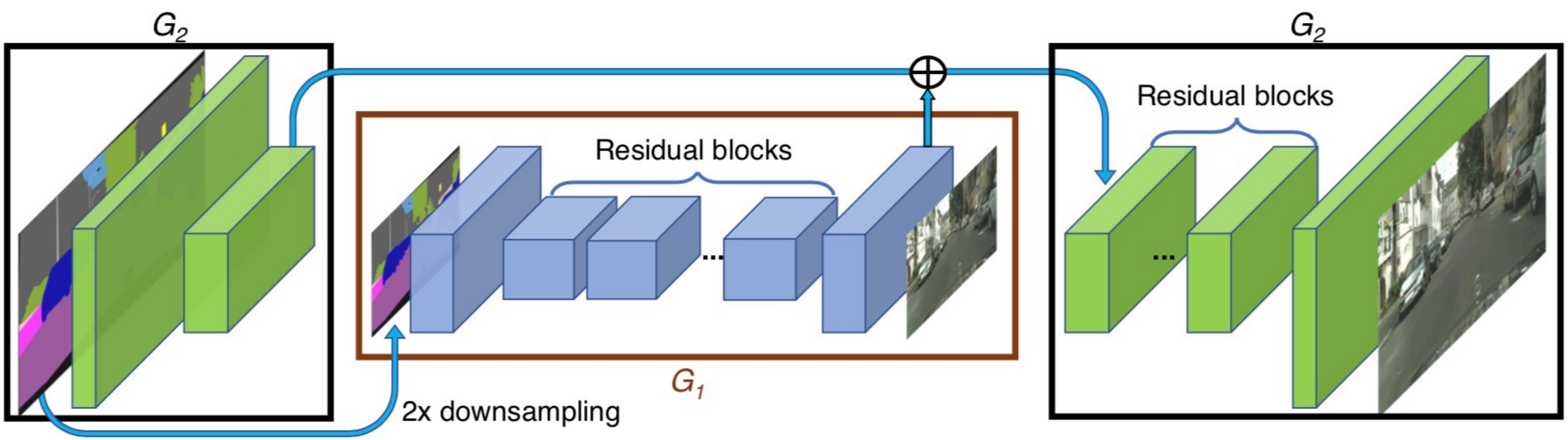
  - Since all distributions above are Gaussian, the KL divergences are tractable

Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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 $\mu_\theta(\mathbf{x}_\mathrm{t}, \mathrm{t})$ shares weights, but conditioned by step t



* Image from the pix2pix-HD paper
Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Diffusion Probabilistic Model

- **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 hundreds of reverse steps (very slow)

    - Early and late steps change the high- and low-level attributes, respectively

* Image from the DDPM paper
Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML'15

Denoising Diffusion Probabilistic Model (DDPM)

- **DDPM reparametrizes the reverse distributions of diffusion models**
  - **Key idea:** The original reverse step fully creates the denoiser $\mu_\theta(\mathbf{x}_\mathrm{t}, \mathrm{t})$ from $\mathbf{x}_t$
    - However, $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ share most information, and thus it is redundant

      → Instead, create the **residual** $\epsilon_\theta(\mathbf{x}_\mathrm{t}, \mathrm{t})$ and add to the original $\mathbf{x}_t$

Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS'20

Denoising Diffusion Probabilistic Model (DDPM)

- **DDPM reparametrizes the reverse distributions of diffusion models**

  - **Key idea:** The original reverse step fully creates the denoiser $\mu_\theta(\mathbf{x}_\mathrm{t}, \mathrm{t})$ from $\mathbf{x}_t$
    - However, $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ share most information, and thus it is redundant

      → Instead, create the **residual** $\epsilon_\theta(\mathbf{x}_\mathrm{t}, \mathrm{t})$ and add to the original $\mathbf{x}_t$

  - Formally, DDPM reparametrizes the learned reverse distribution as[1]

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

  and the step-wise objective L_{t-1} can be reformulated as[2]

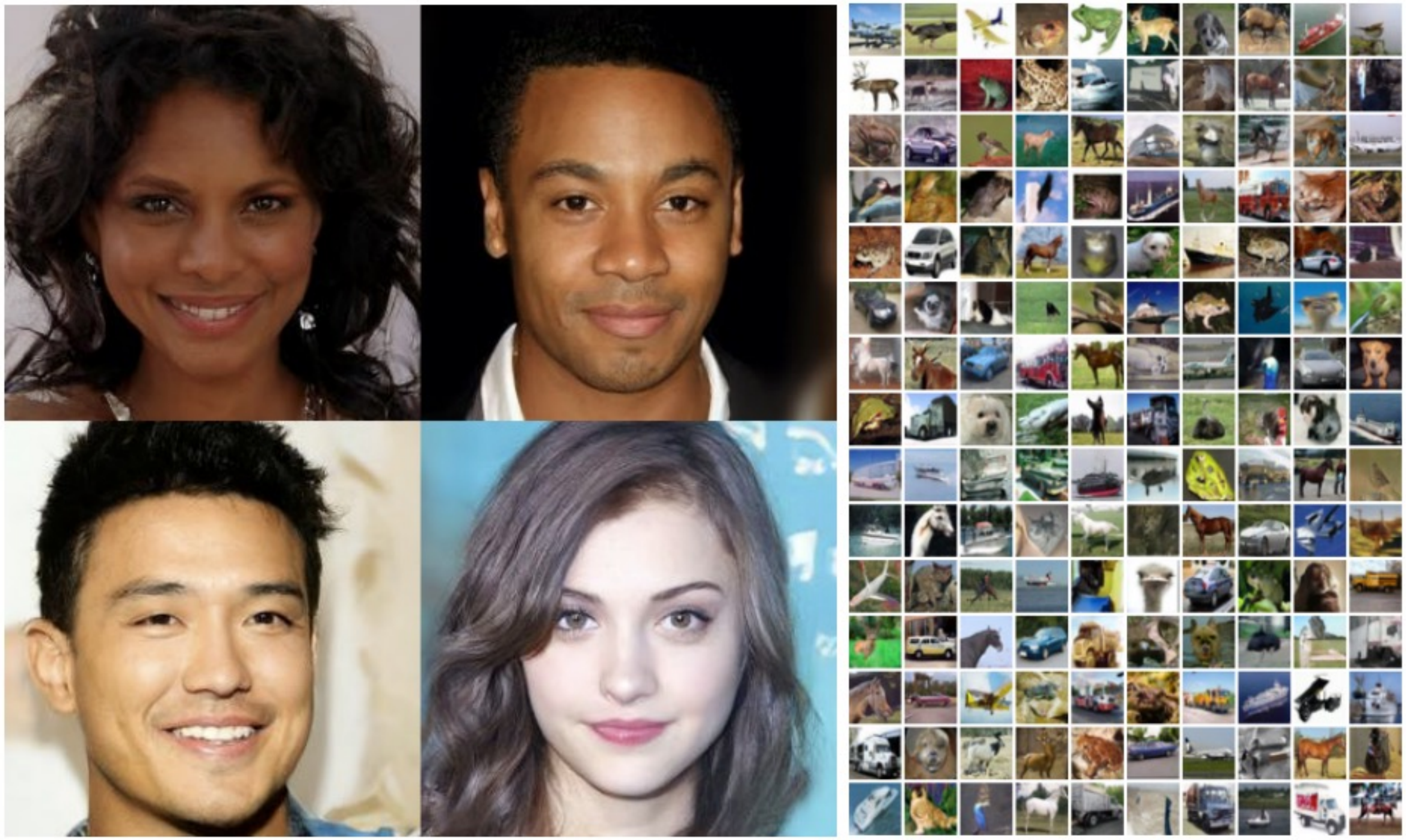
$$\mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

1. α_t are some constants determined by β_t
2. Note that we need no “intermediate” samples, and only compare the forward noise $\boldsymbol{\epsilon}$ and reverse noise $\boldsymbol{\epsilon}_\theta$ conditioned on $\mathbf{x}_0$

Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS'20

Denoising Diffusion Probabilistic Model (DDPM)

- **DDPM initiated the diffusion model boom**
  - Achieved SOTA on CIFAR-10, with high-resolution scalability
  - It produces more diverse samples than GAN (no mode collapse)



Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS'20

Denoising Diffusion Implicit Model (DDIM)

- **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

Song et al. Denoising Diffusion Implicit Models. ICLR'21

Denoising Diffusion Implicit Model (DDIM)

- **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 $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$[1]
    - Unlike original diffusion model, it is not a Markovian structure

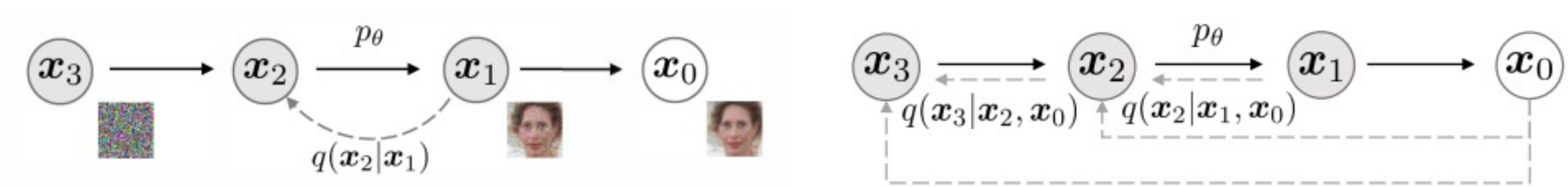


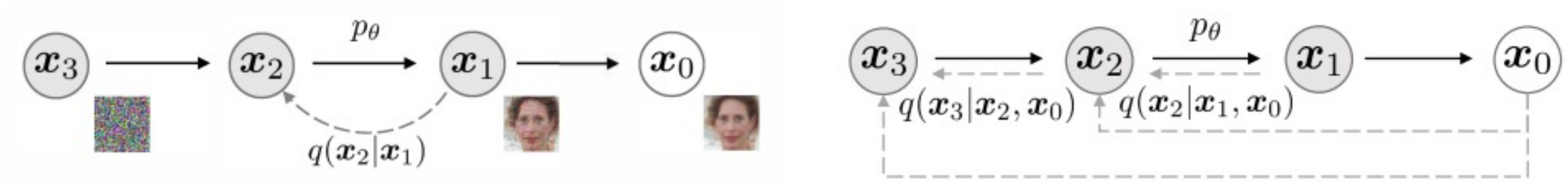
Figure 1: Graphical models for diffusion (left) and non-Markovian (right) inference models.

1. Recall that the original diffusion model uses $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

Song et al. Denoising Diffusion Implicit Models. ICLR'21

Denoising Diffusion Implicit Model (DDIM)

- **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 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$



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

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

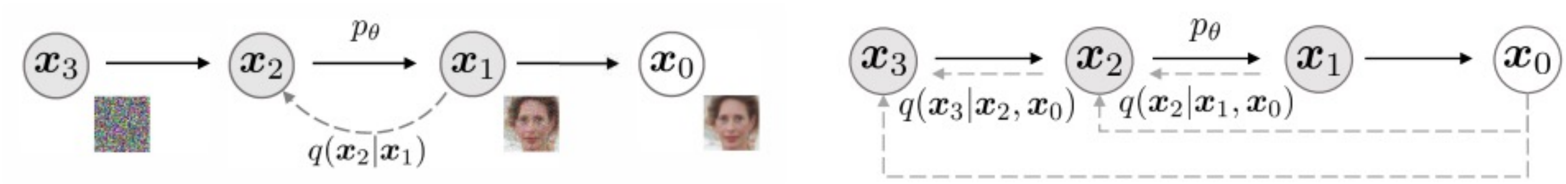
then, the forward process is derived from Bayes' rule

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

Song et al. Denoising Diffusion Implicit Models. ICLR'21

Denoising Diffusion Implicit Model (DDIM)

- **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 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$



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

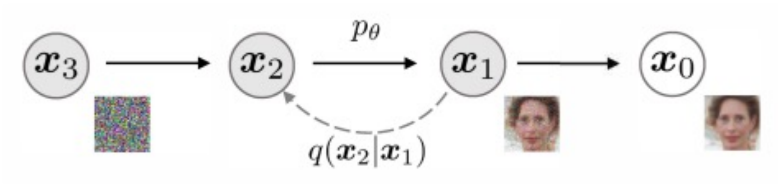
    and reverse process is

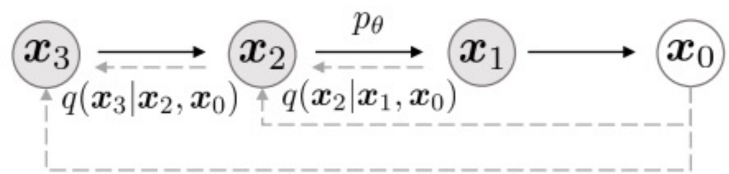
$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{“ predicted } \boldsymbol{x}_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{“direction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

Song et al. Denoising Diffusion Implicit Models. ICLR’21

Denoising Diffusion Implicit Model (DDIM)

- **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 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$





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

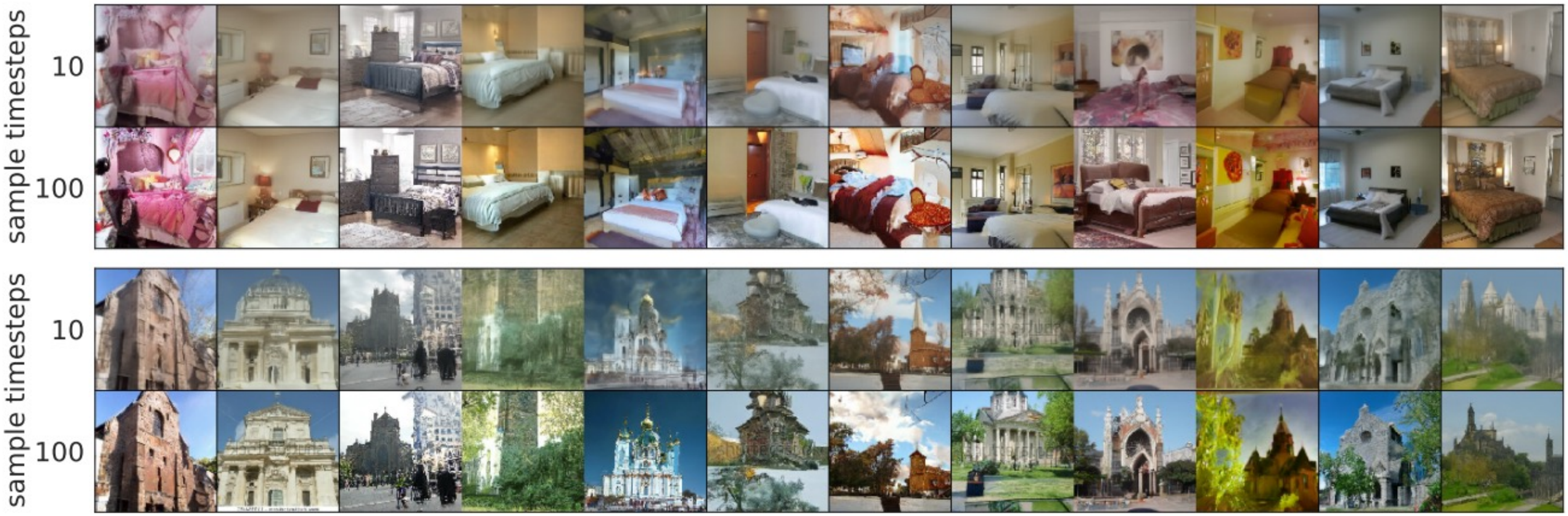
  - **Training:** The variational lower bound of DDIM is identical to the one of DDPM[1]
    - It is surprising since the forward/reverse formulation is totally different

1. Precisely, the bound is different, but the solution is identical under some assumption (though violated in practice)

Song et al. Denoising Diffusion Implicit Models. ICLR’21

Denoising Diffusion Implicit Model (DDIM)

- **DDIM significantly reduces the sampling steps of diffusion model**
  - Creates the outline of the sample after only 10 steps (DDPM needs hundreds)

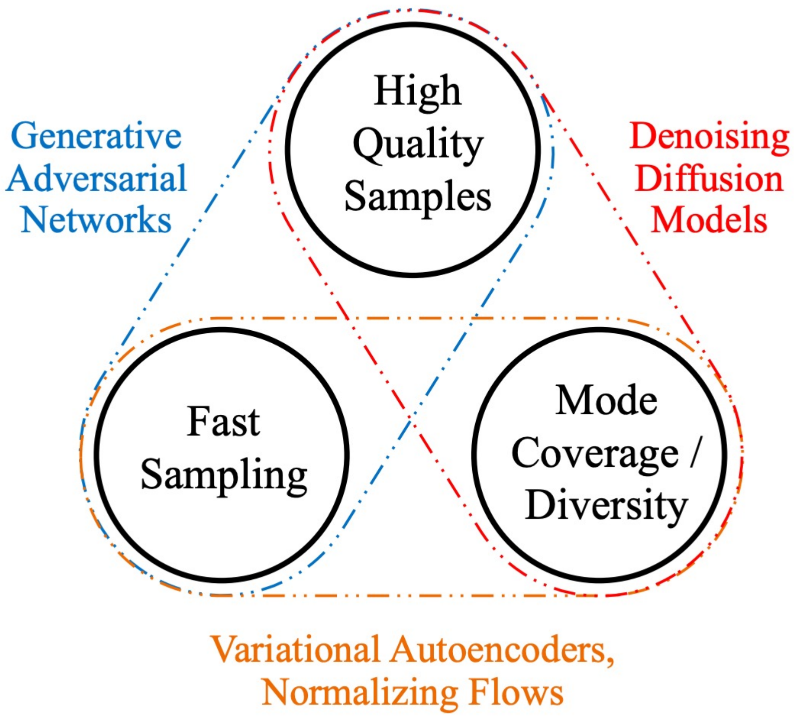


Song et al. Denoising Diffusion Implicit Models. ICLR'21

Take-home Message

- **New golden era of generative models**
  - Competition of various approaches: GAN, VAE, flow, diffusion model[1]
  - Also, lots of hybrid approaches (e.g., score SDE = diffusion + continuous flow)

- **Which model to use?**
  - **Diffusion model** seems to be a nice option for high-quality generation
  - However, **GAN** is (currently) still a more practical solution which needs fast sampling (e.g., real-time apps.)



1. VAE also shows promising generation performance (see NVAE, very deep VAE)

Thank you for listening! 😀