Lecture 4: Generative Diffusion Models

Announcements

- Paper bidding ended
  - Great thanks for all students who submitted response on time!
- Paper assignment and presentation schedule is released
  - Great thanks for our GSI, Yixuan!
  - Check course schedule
  - If you are satisfied with the paper and time slot you were assigned to, please sign your name on this spreadsheet by 11:59 PM Sep 14
- Paper presentation will start from next week
  - Get prepared for live evaluation of each presentation on the class!
  - Voted by all students and instructors from two aspects:
    - Presentation content: score 1-5 (clarity of message, depth of information, key points, etc.)
    - Presentation format: score 1-5 (slide layout, organization, visual design, interaction, etc.)

Announcements

- Assignment submission:
  - Paper presentation slides:
    - One-time submission on Canvas by 11:59 PM on the Friday before your presentation
  - Paper review:
    - Weekly submission on Canvas by 11:59 PM on Monday and Wednesday before paper presentation
    - Weekly submission on Piazza to earn your bonus point!
      - Submit your review to Piazza anytime
      - All students and instructors vote for top reviews by clicking "helpful"
      - One outstanding reviewer each paper
        - Get the most votes before submission due time
        - If the same votes, go for earlier post on Piazza
      - 4 outstanding reviewers each week get +1 bonus points

Announcements

- Suggestion and feedback for the class:
  - More technical details …
  - Any more suggestions?

Today's agenda

- Generative diffusion models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

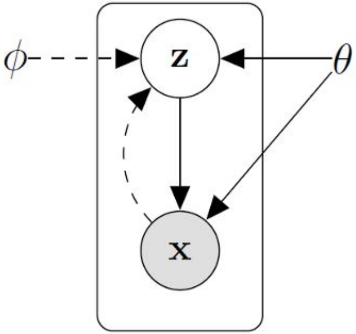
**Some slides in this section are adapted from the post:* *[Generative Modeling by Estimating Gradients of the Data Distribution]*

Generative models

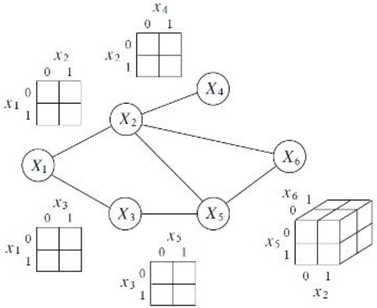
- Goal of generative modeling:
  - Fit a model to the data distribution such that we can synthesize new data points at will by sampling from the distribution

Generative models

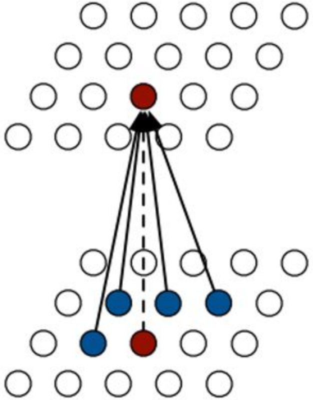
- Likelihood-based generative models
  - Directly learn the distribution’s probability density function via (approximate) maximum likelihood
  - Strong restrictions on the model architecture to ensure a tractable normalizing constant for likelihood computation



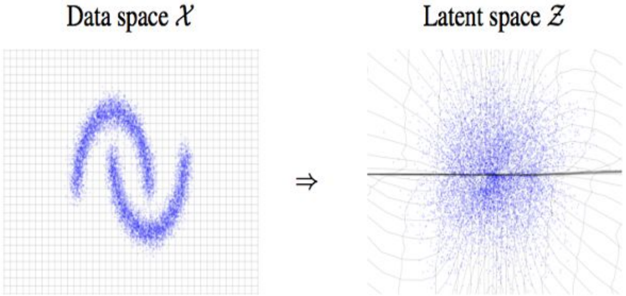
Bayesian networks (e.g., VAEs)



MRF



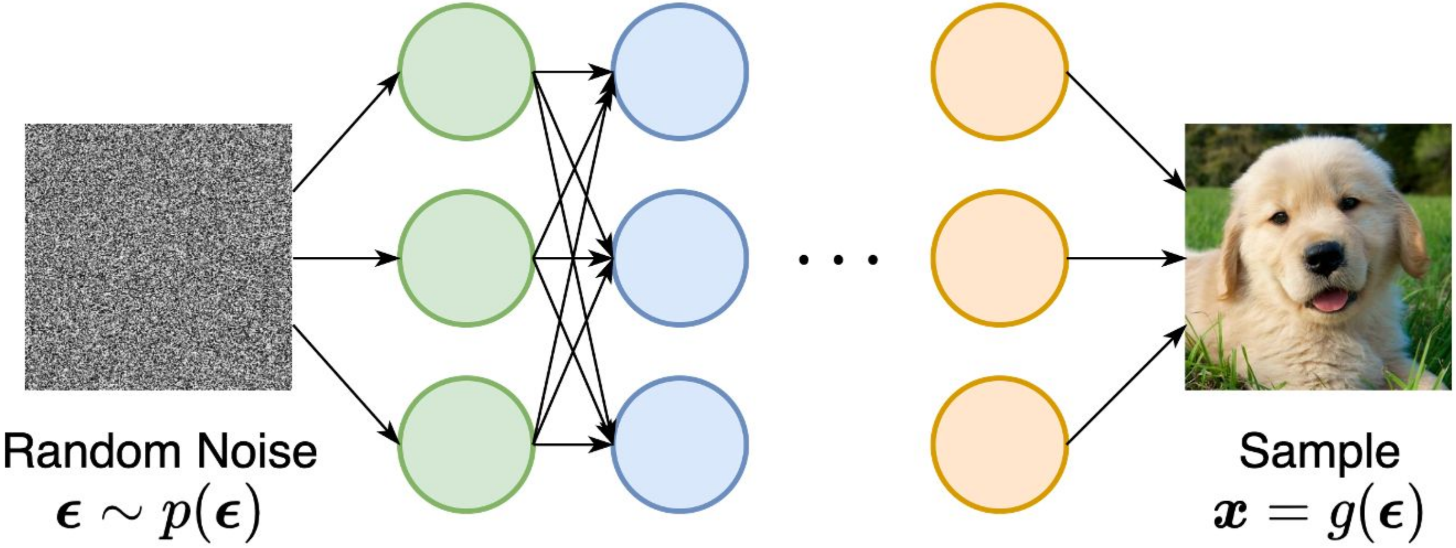
Autoregressive models



Flow models

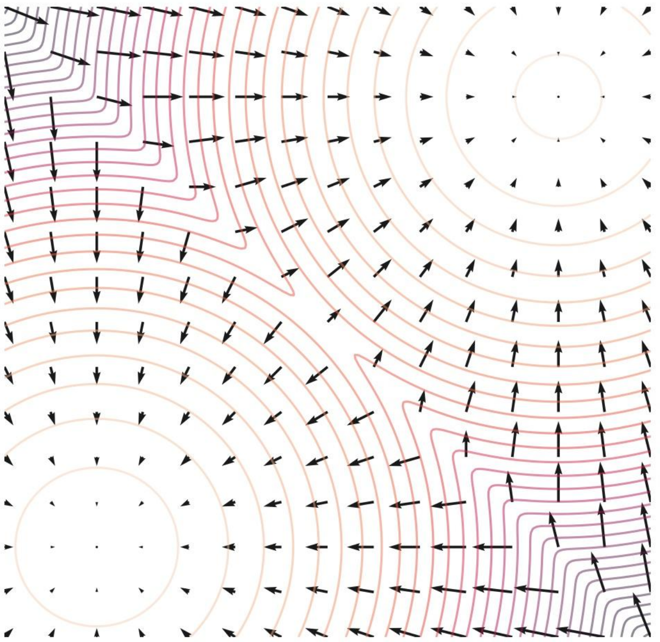
Generative models

- Implicit generative models
  - Probability distribution is implicitly represented by a model of its sampling process
  - Require adversarial training, which is notoriously unstable and can lead to mode collapse



Generative models

- Score-based generative models
  - Score function:
    - The gradient of the log probability density function
  - Not required to have a tractable normalizing constant
  - Can be directly learned by score matching



Score-based generative models

- Goal of generative modeling:
  - Fit a model to the data distribution such that we can synthesize new data points at will by sampling from the distribution
  - Suppose we are given a dataset $\{x_1, x_2, \ldots, x_N\}$, where each point is drawn independently from an underlying data distribution p(x)
  - Model the probability density function (PDF) as:

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta} \qquad \int p_\theta(\mathbf{x})\mathrm{d}\mathbf{x} = 1. \qquad \max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i).$$

  - $f_\theta(x)$: unnormalized probabilistic model or energy-based model, parameterized by θ
  - Z_θ: normalizing constant - Always intractable!

Score-based generative models

- Score function of a distribution p(x): $\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p(\mathbf{x})$
- Score-based model $s_\theta(x)$:

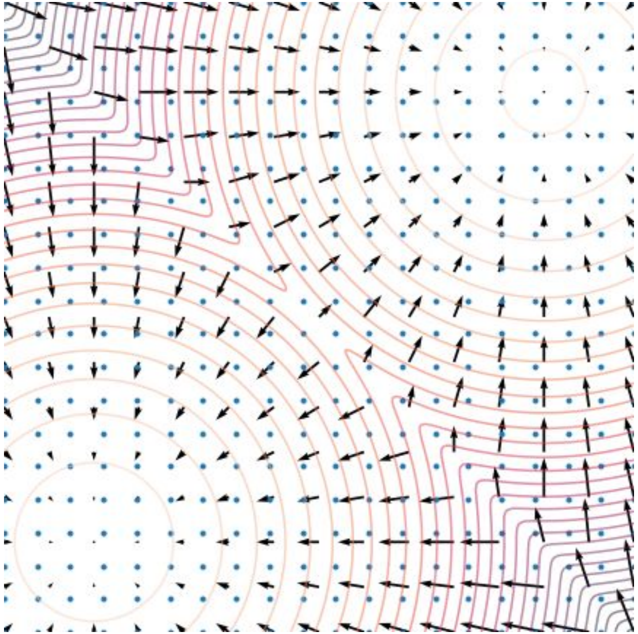
$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$$

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z_\theta}_{=0} = -\nabla_\mathbf{x} f_\theta(\mathbf{x})$$

- Score-based model $s_\theta(x)$ is independent of the normalizing constant Z_θ !

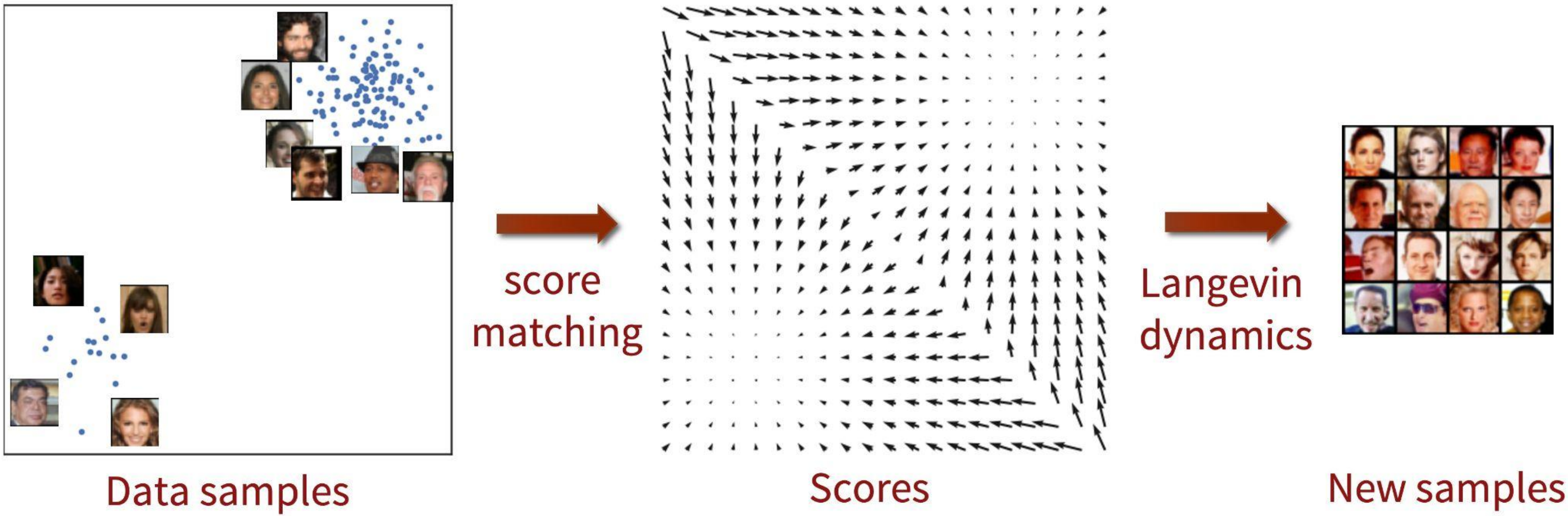
Score-based generative models

- Langevin dynamics
  - Iterative procedure to sample from distribution p(x) using only its score function
  - When ϵ is sufficiently small and K is sufficiently large:



Score-based generative models

- Naive score-based generative modeling

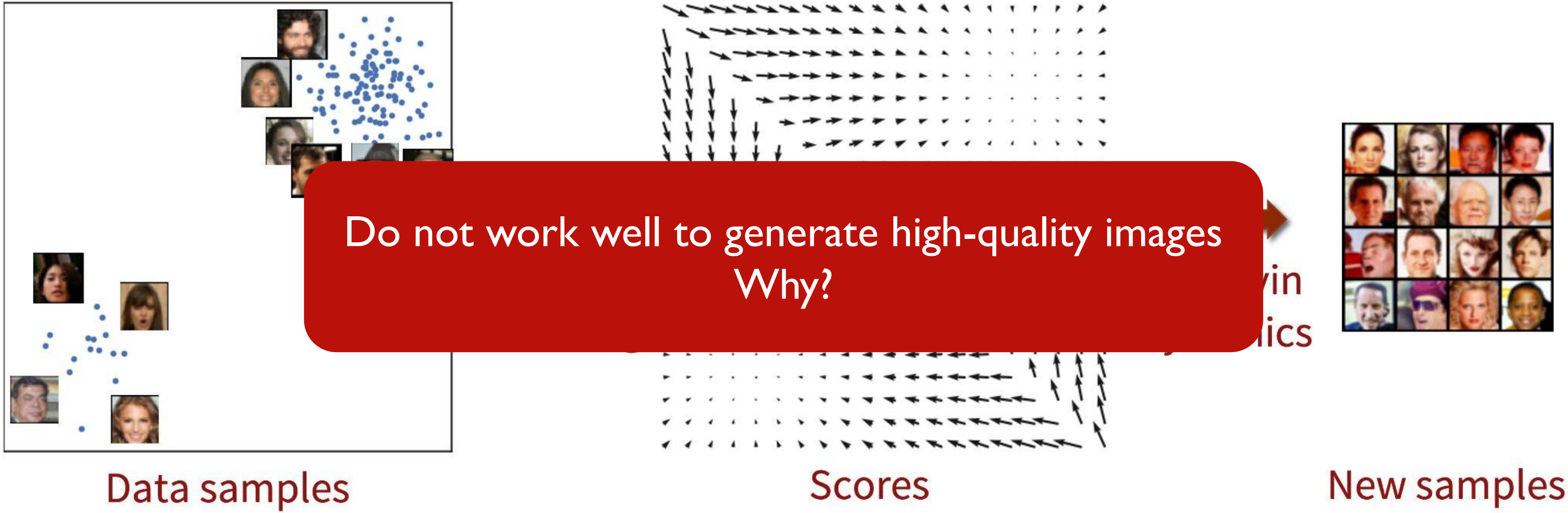


$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p(\mathbf{x})$$

Score-based generative models

- Naive score-based generative modeling



Do not work well to generate high-quality images
Why?

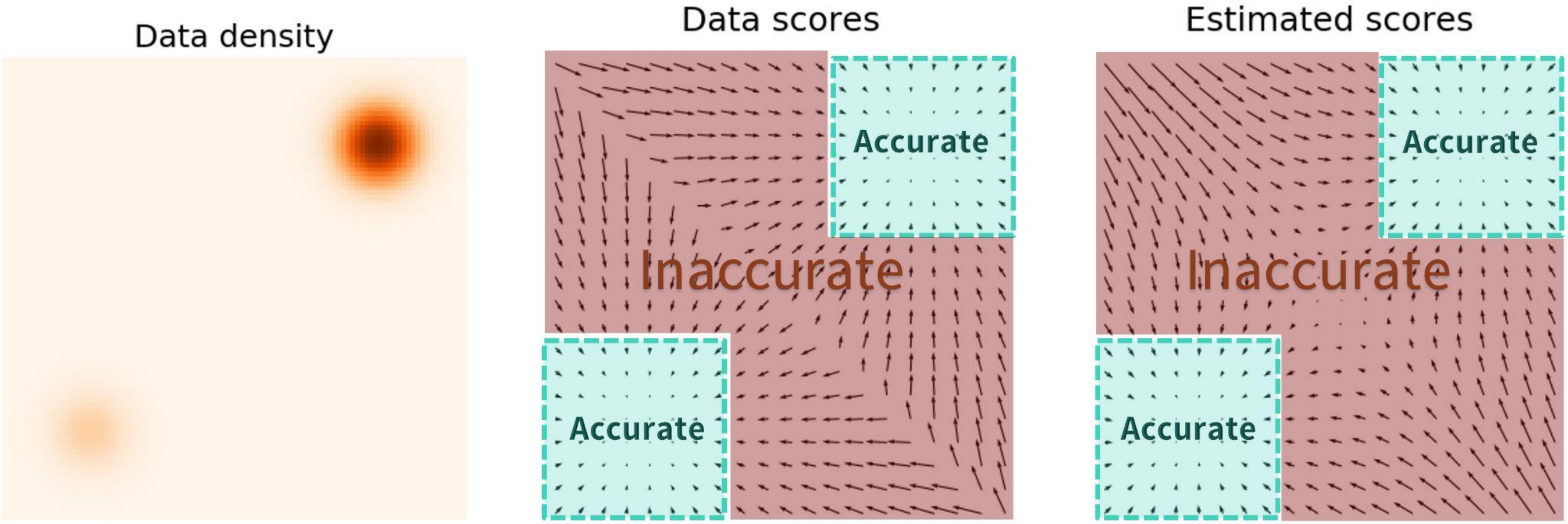
$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p(\mathbf{x})$$

Score-based generative models

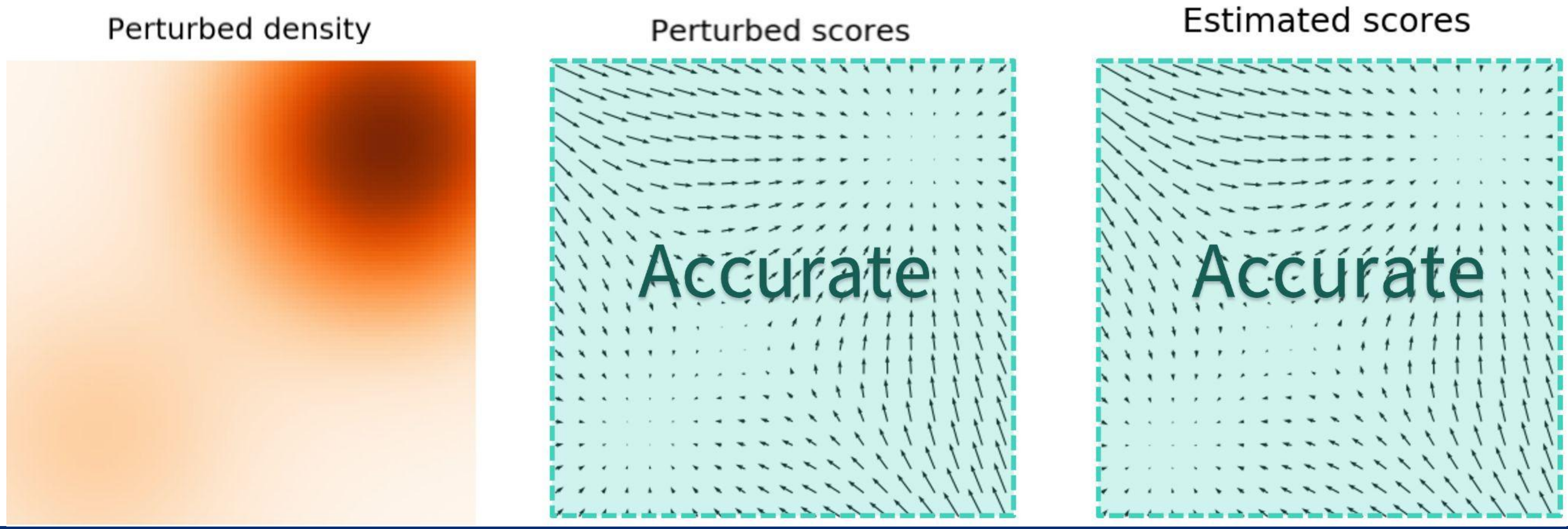
- Naive score-based generative modeling
  - Estimated score functions are inaccurate in low density regions, where few data points are available for computing the score matching objective

$$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2] = \int p(\mathbf{x})\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2 \mathrm{d}\mathbf{x}.$$



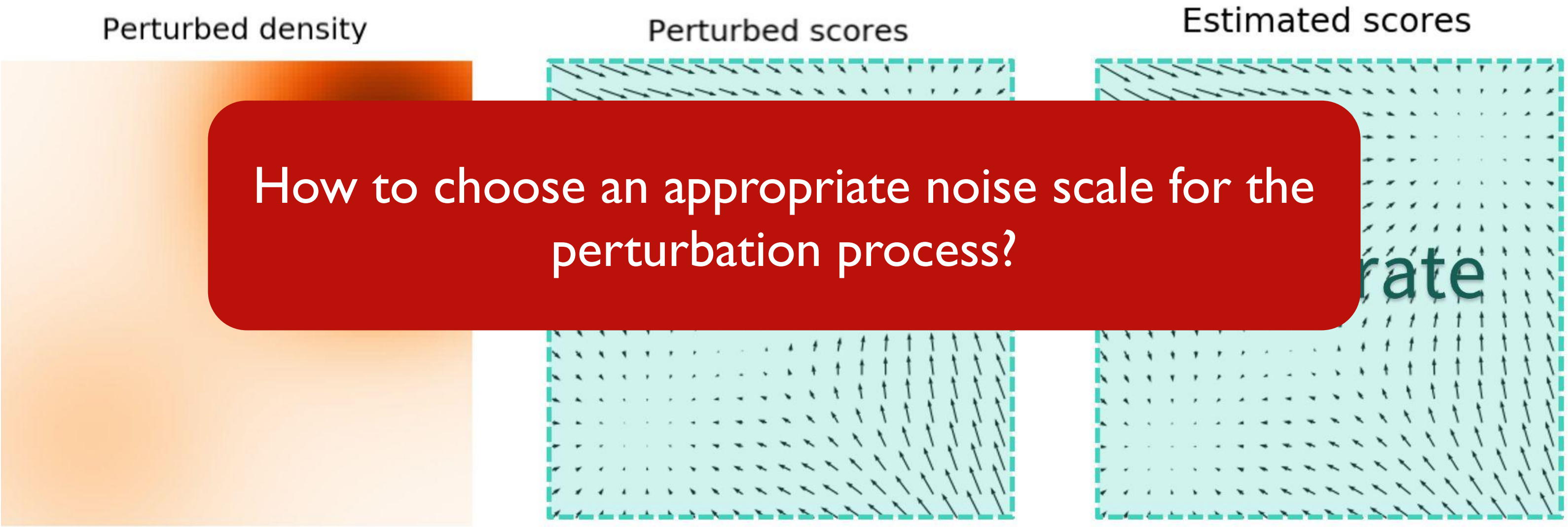
Score-based generative models

- Solution:
  - Perturb data points with noise and train score-based models on the noisy data points
  - When the noise magnitude is sufficiently large, it can populate low data density regions to improve the accuracy of estimated scores



Score-based generative models

- Solution:
  - Perturb data points with noise and train score-based models on the noisy data points
  - When the noise magnitude is sufficiently large, it can populate low data density regions to improve the accuracy of estimated scores



How to choose an appropriate noise scale for the perturbation process?

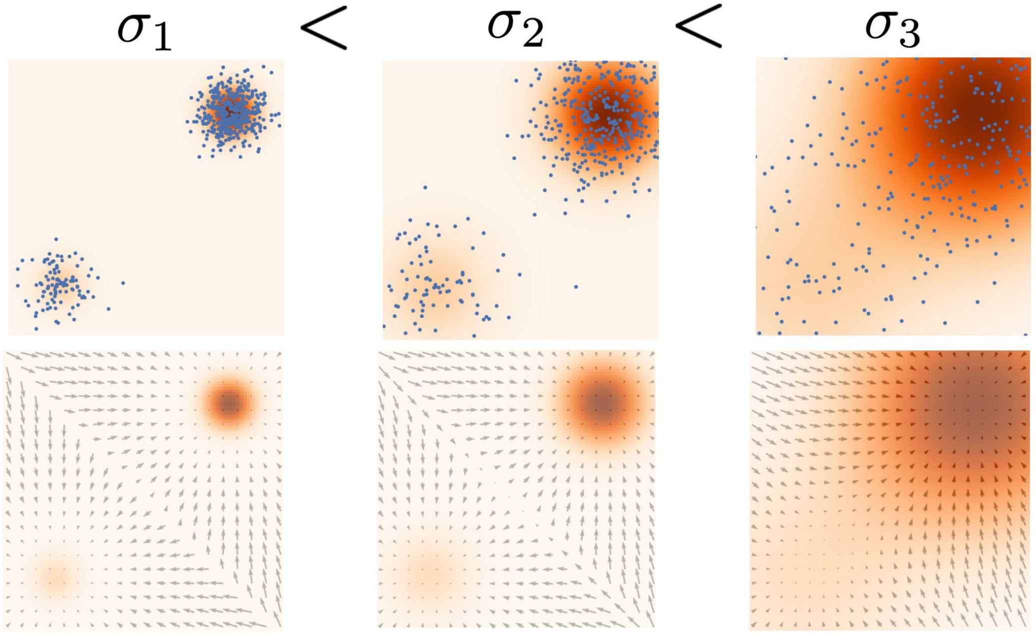
Score-based generative models

- Multiple scales of noise perturbations:
  - Let there be a total of L increasing standard deviations: $\sigma_1 < \sigma_2 < \cdots < \sigma_L$.
  - Perturb the data distribution p(x) with each of the Gaussian noise: $\mathcal{N}(0, \sigma_i^2 I), i = 1, 2, \cdots, L$
  - Noise-perturbed distribution:

$$p_{\sigma_i}(\mathbf{x}) = \int p(\mathbf{y}) \mathcal{N}(\mathbf{x}; \mathbf{y}, \sigma_i^2 I) \mathrm{d}\mathbf{y}$$

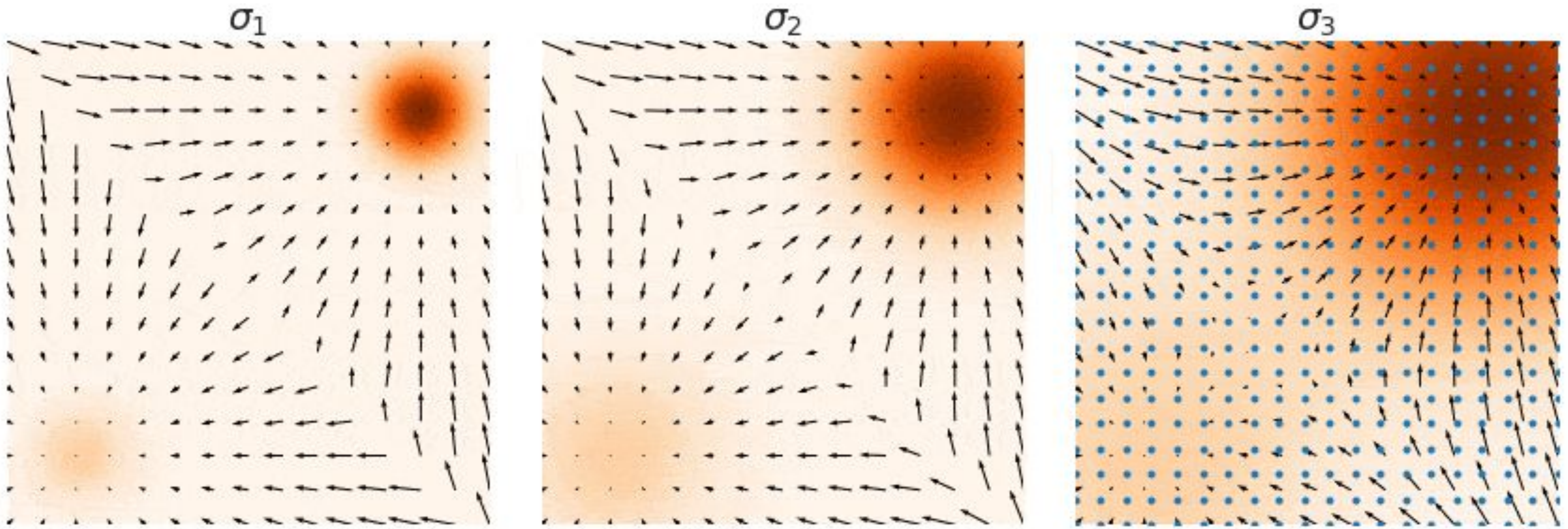
Score-based generative models

- Score function of each noise-perturbed distribution: $\mathbf{s}_\theta(\mathbf{x}, i) \approx \nabla_\mathbf{x} \log p_{\sigma_i}(\mathbf{x})$
- Noise Conditional Score-Based Model:
  - Noise Conditional Score Network (NCSN)



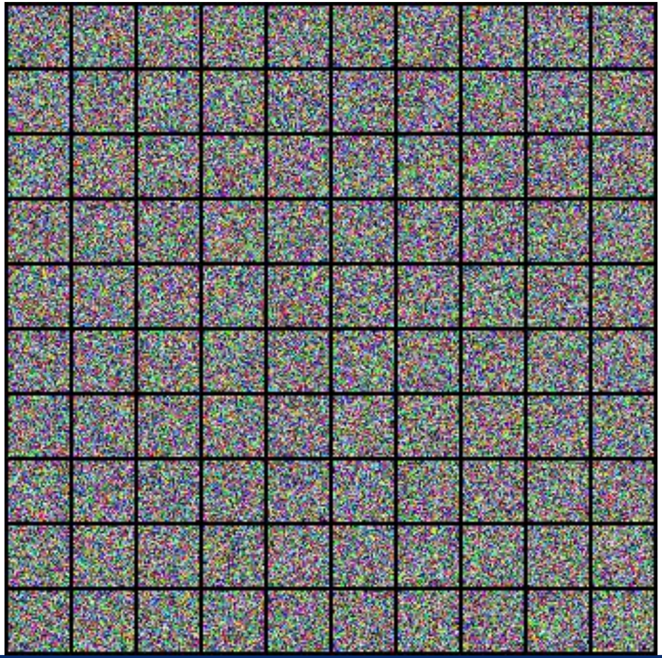
Score-based generative models

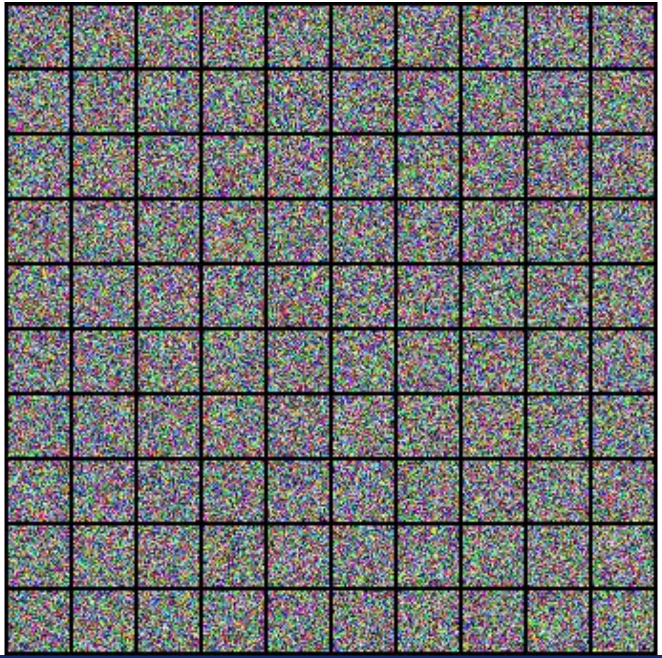
- Sample from noise-conditional score-based model:
  - Annealed Langevin dynamics
  - Run Langevin dynamics for i = L, L-1, …, 1 in sequence with decreasing noise scales



Score-based generative models

- Sample from noise-conditional score-based model:
  - Annealed Langevin dynamics
  - Run Langevin dynamics for i = L, L-1, …, 1 in sequence with decreasing noise scales

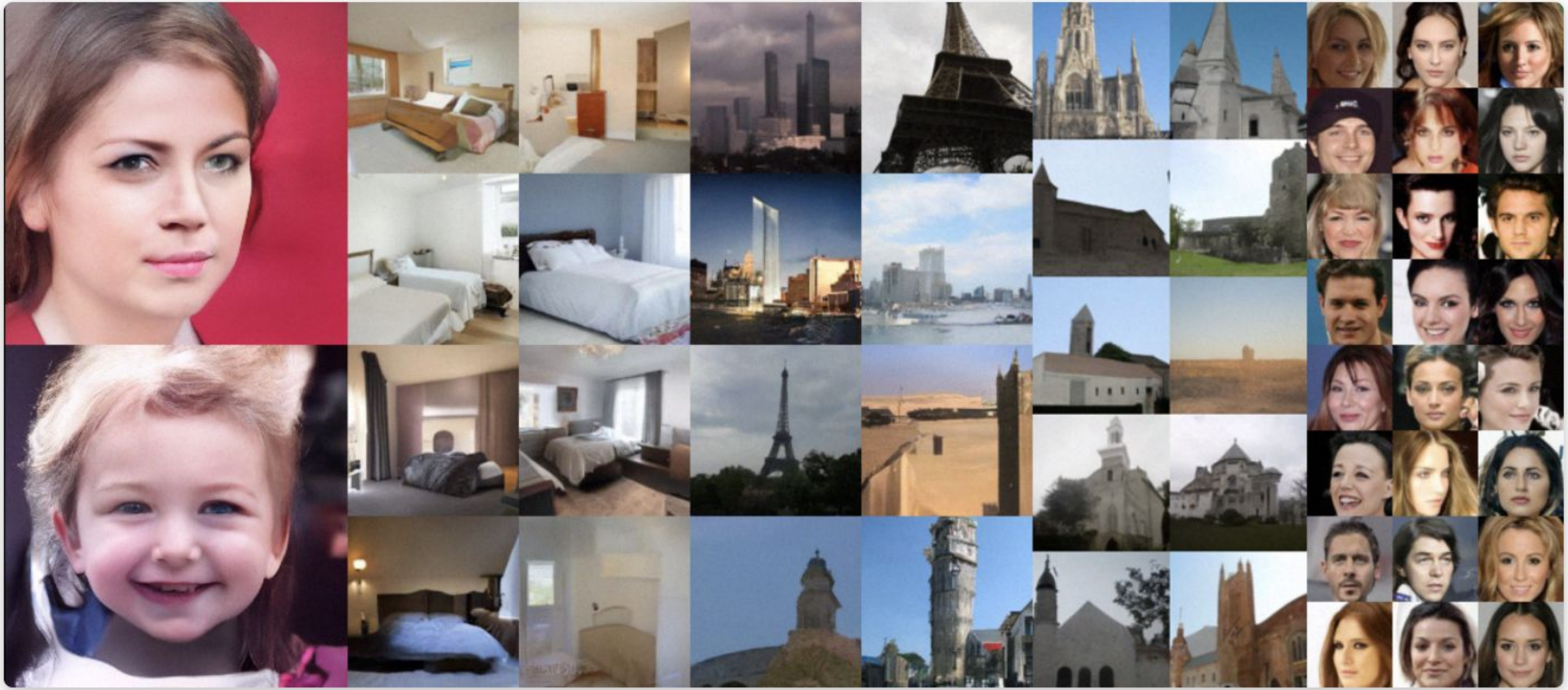




Score-based generative models

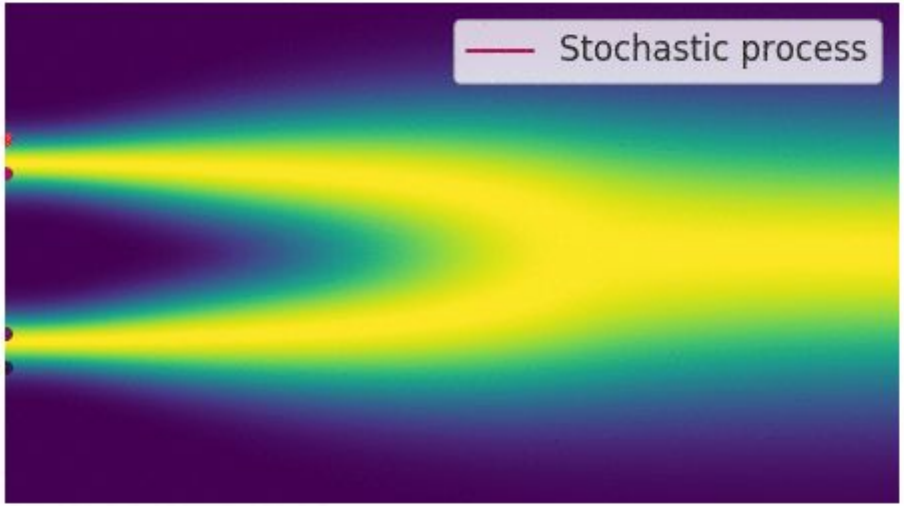
- NCSNv2 model:

Generate high-quality images comparable to GAN models!



Score-based generative models

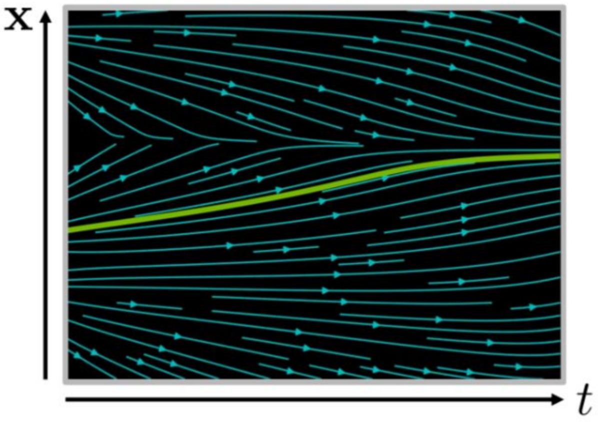
- Stochastic differential equations (SDEs)
  - When the number of noise scales approaches infinity, we essentially perturb the data distribution with continuously growing levels of noise
  - Noise perturbation procedure is a continuous-time stochastic process



Score-based generative models

Ordinary Differential Equation (ODE):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, t) \quad \text{or} \quad \mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t$$



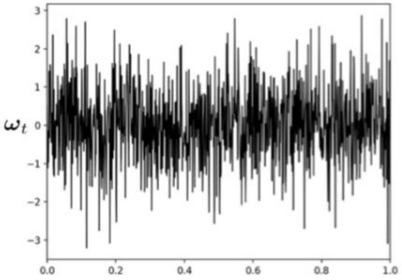
Analytical Solution: $$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}(\mathbf{x}, \tau)\mathrm{d}\tau$$

Iterative Numerical Solution: $$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), t)\Delta t$$

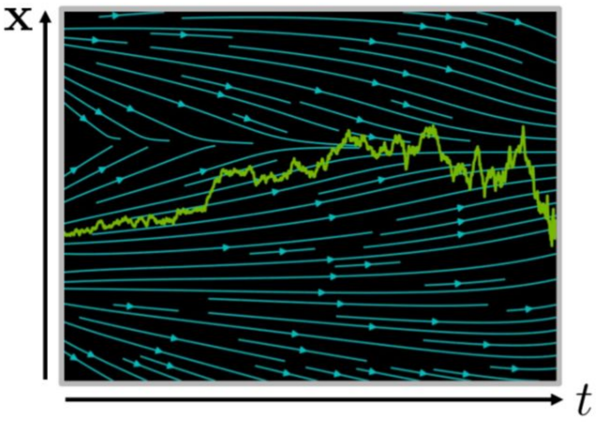
Stochastic Differential Equation (SDE):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \underbrace{\mathbf{f}(\mathbf{x}, t)}_{\text{drift coefficient}} + \underbrace{\sigma(\mathbf{x}, t)}_{\text{diffusion coefficient}}\boldsymbol{\omega}_t$$

Wiener Process (Gaussian White Noise)



$$\left(\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + \sigma(\mathbf{x}, t)\mathrm{d}\boldsymbol{\omega}_t \right)$$



$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), t)\Delta t + \sigma(\mathbf{x}(t), t)\sqrt{\Delta t}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$$

Score-based generative models

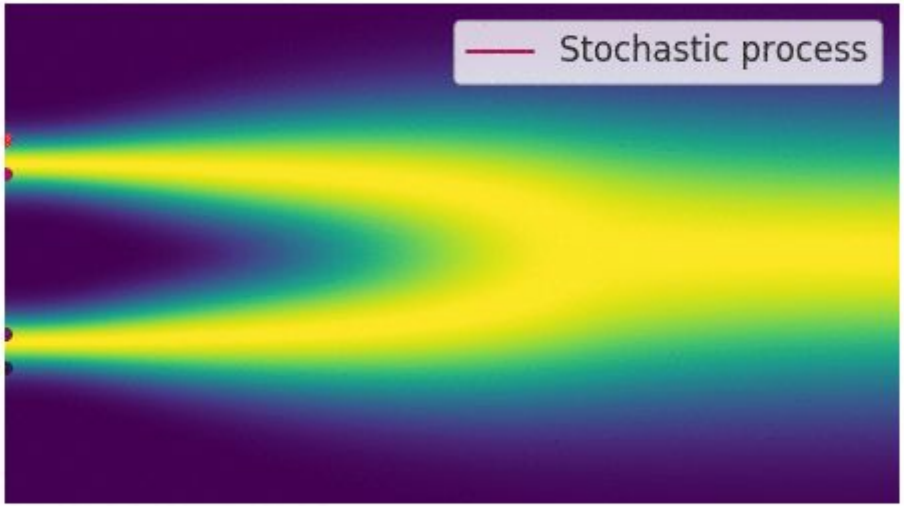
- Stochastic differential equations (SDEs)
  - Stochastic processes (diffusion processes) are solutions of stochastic differential equations (SDEs):

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

  - f(, t): a vector-valued function called the drift coefficient
  - g(t): a real-valued function called the diffusion coefficient
  - w: a standard Brownian motion
  - dw: can be viewed as infinitesimal white noise

Score-based generative models

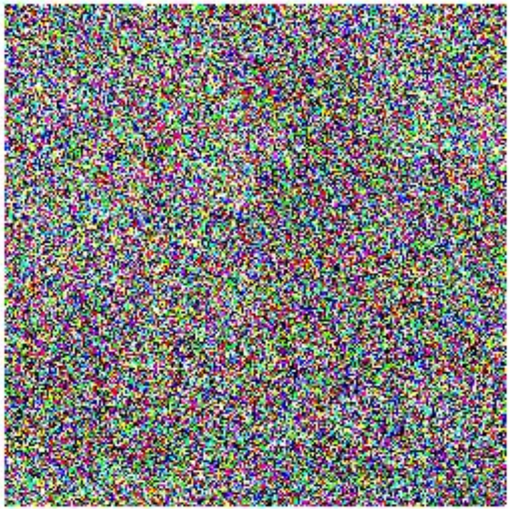
- Perturbing data with an Stochastic differential equation (SDE)

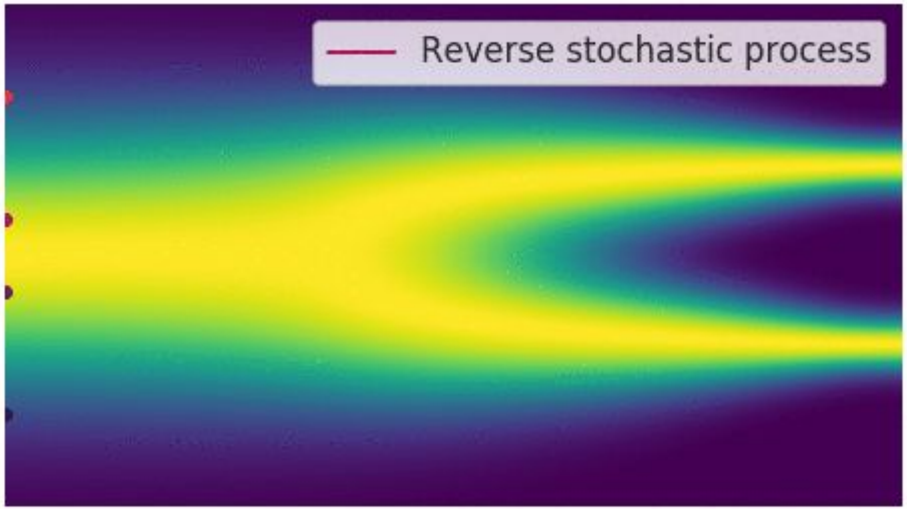


Score-based generative models

- Reversing the SDE for sample generation
  - Need to estimate the score function $p_t(x)$

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}]dt + g(t)d\mathbf{w}$$

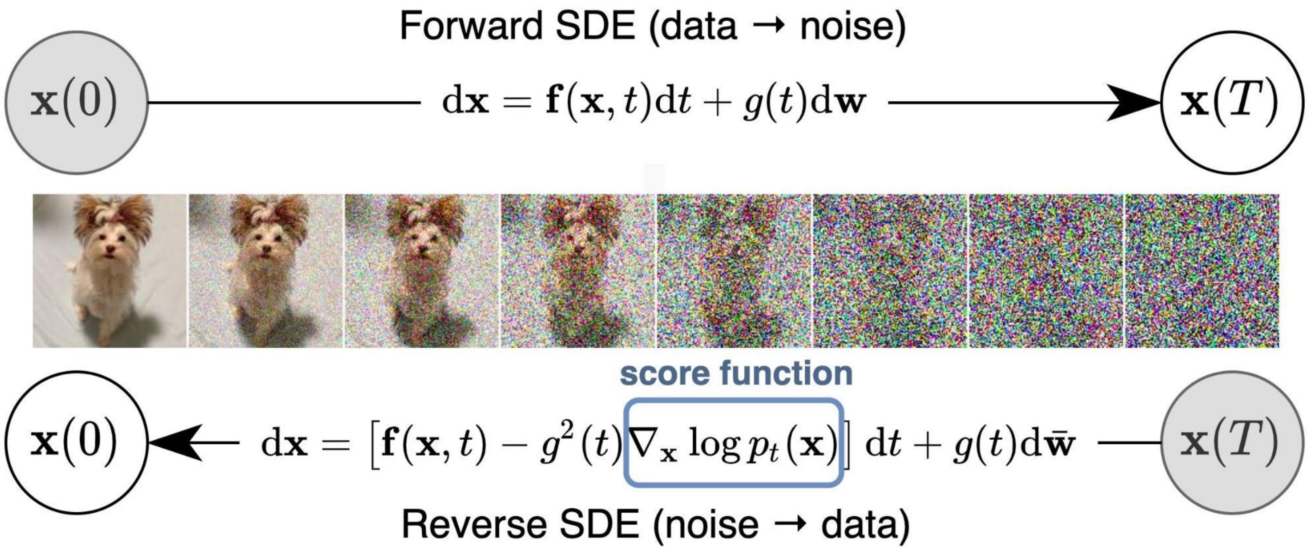




Score-based generative models

- Reversing the SDE for sample generation
  - Need to estimate the score function $p_t(x)$

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}]dt + g(t)d\mathbf{w}$$



Score-based generative models

- Reversing the SDE for sample generation
  - Need to estimate the score function $p_t(x)$

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

  - Train a time-dependent score-based model

$$\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$$

$$\mathbb{E}_{t \in \mathcal{U}(0,T)} \mathbb{E}_{p_t(\mathbf{x})}[\lambda(t) \|\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, t)\|_2^2].$$

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\mathbf{s}_\theta(\mathbf{x}, t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

Score-based generative models

- Reversing the SDE for sample generation
  - Need to estimate the score function $p_t(x)$

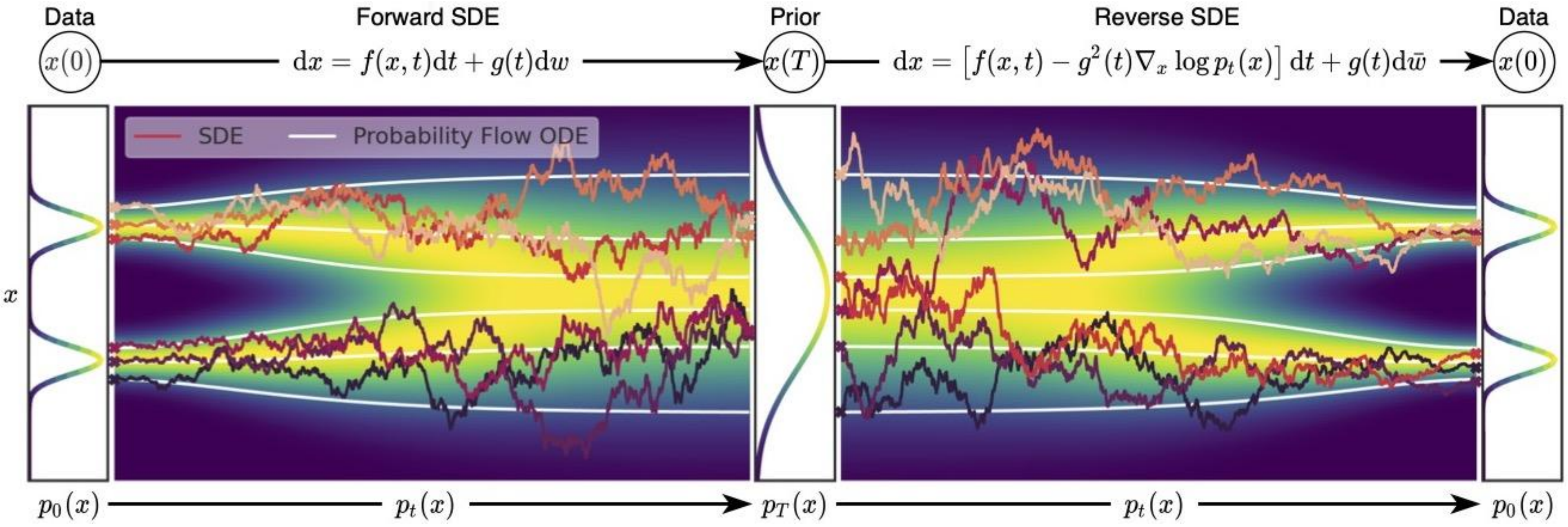
$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}]dt + g(t)d\mathbf{w}$$

  - Train a time-dependent score-based model
  - Simulate reverse stochastic process for sample generation:
    - Numerical SDE solver, such as Euler-Maruyama method

$$\Delta\mathbf{x} \leftarrow [\mathbf{f}(\mathbf{x}, t) - g^2(t)\mathbf{s}_\theta(\mathbf{x}, t)]\Delta t + g(t)\sqrt{|\Delta t|}\mathbf{z}_t$$
$$\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}$$
$$t \leftarrow t + \Delta t,$$

Score-based generative models

- Map data to a noise distribution (the prior) with an SDE
- Reverse the SDE for generative modeling



Today's agenda

- Generative diffusion models
    - Score-based generative models
    - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

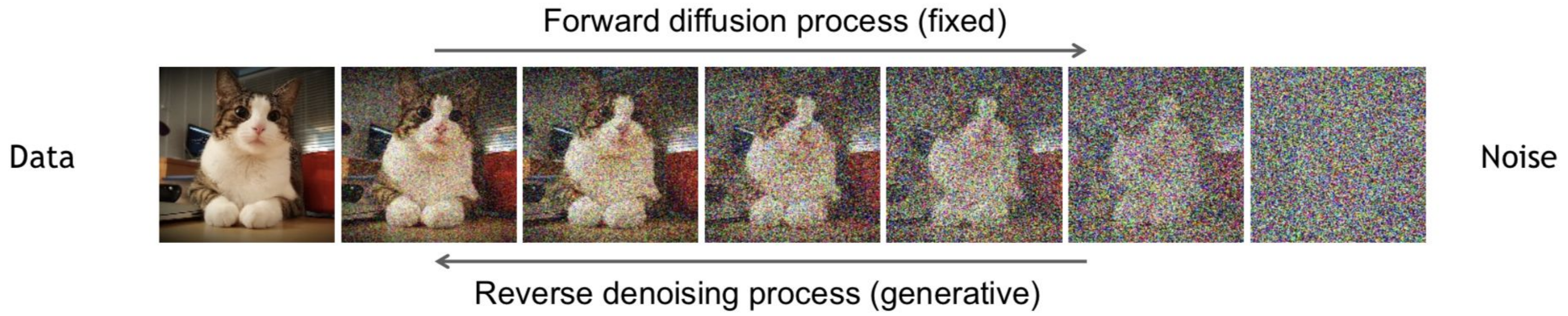
Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

**Some slides in this section are adapted from: [CVPR 2022 Tutorial: Denoising Diffusion-based Generative Modeling: Foundations and Applications]*

Denoising diffusion probabilistic models

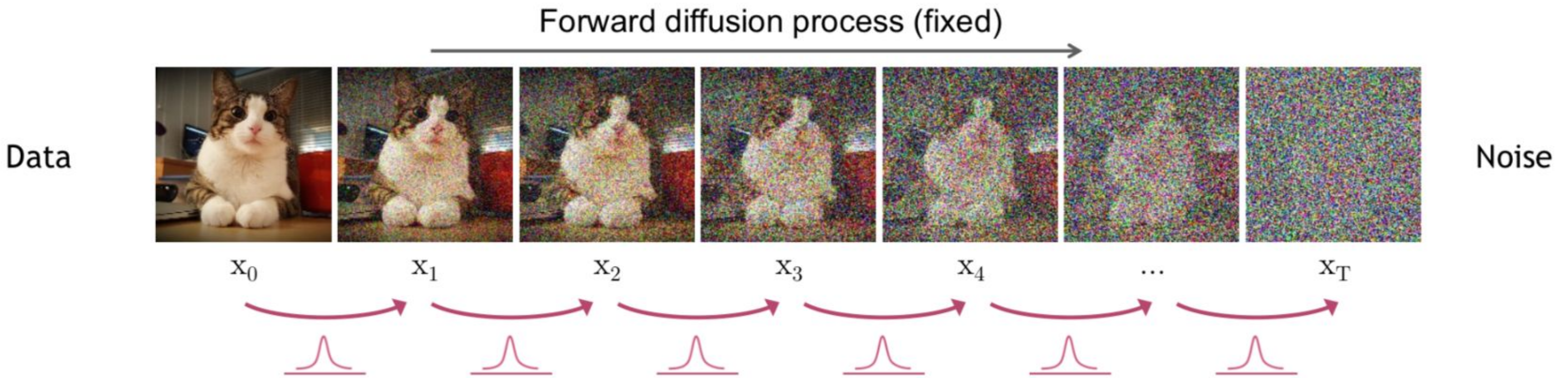
- Denoising diffusion models consist of two processes:
  - Forward diffusion process that gradually adds noise to input
  - Reverse denoising process that learns to generate data by denoising



Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

Denoising diffusion probabilistic models

- Formal definition of the forward process in T steps:



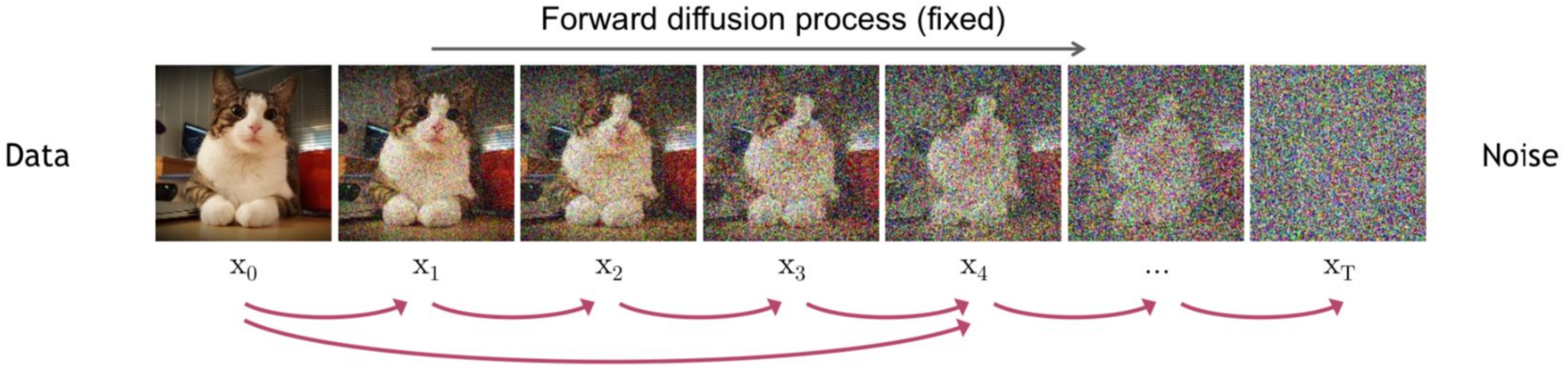
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(joint)

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

Denoising diffusion probabilistic models

- Formal definition of the forward process in T steps:



Define $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ ➡ $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}))$ (Diffusion Kernel)

For sampling: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

β_t values schedule (i.e., the noise schedule) is designed such that $\bar{\alpha}_T \rightarrow 0$ and $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}))$

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

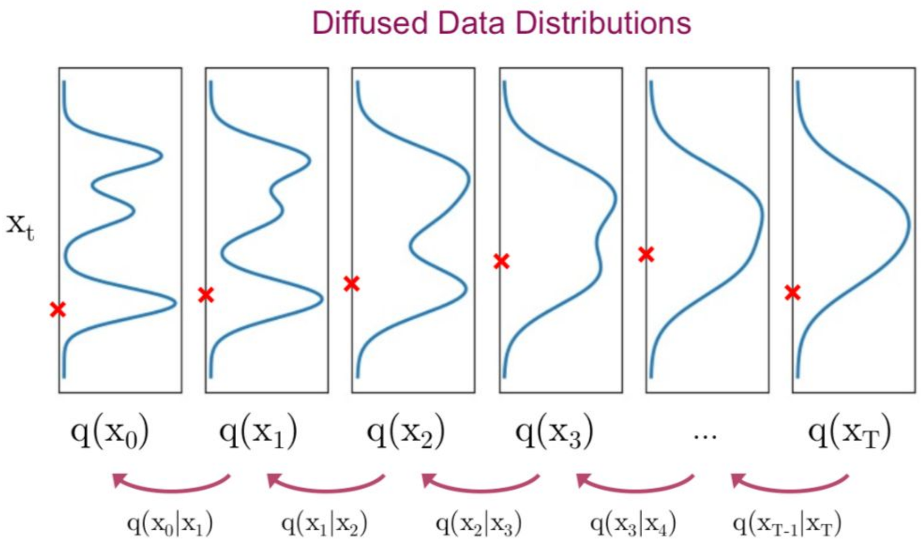
Denoising diffusion probabilistic models

- Generative learning by denoising:

Generation:

Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

Iteratively sample $\mathbf{x}_{t-1} \sim \underbrace{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}_{\text{True Denoising Dist.}}$



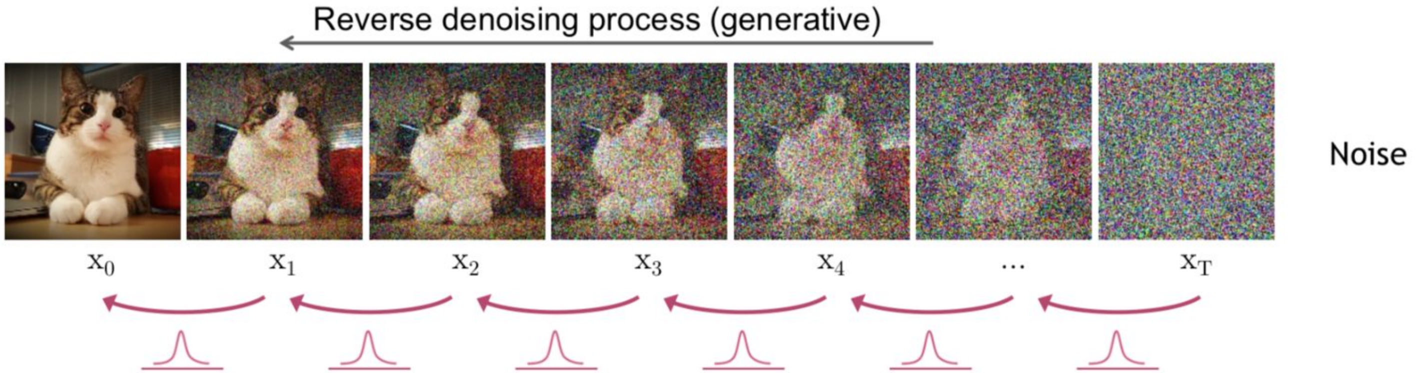
In general, $q(\mathbf{x}_{t-1}|\mathbf{x}_t) \propto q(\mathbf{x}_{t-1})q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is intractable.

Can we approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$? Yes, we can use a Normal distribution if β_t is small in each forward diffusion step.

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

Denoising diffusion probabilistic models

- Generative learning by denoising:



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\text{Trainable network (U-net, Denoising Autoencoder)}}, \sigma_t^2 \mathbf{I})$$

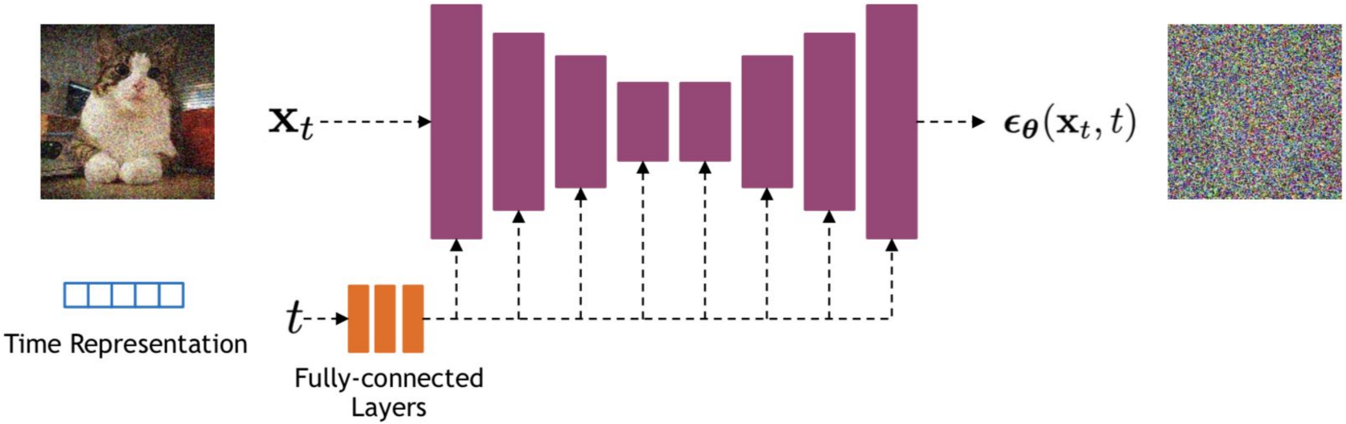
$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

Denoising diffusion probabilistic models

- Propose to represent mean of denoising model using a noise-prediction network

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\right)$$



Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

Denoising diffusion probabilistic models

- Propose to represent mean of denoising model using a noise-prediction network:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\right)$$

- Objective for training diffusion model:
  - Can be rewritten as a combination of several KL-divergence and entropy terms
  - See the detailed step-by-step process in Appendix B in Sohl-Dickstein et al., 2015

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

Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020.

Denoising diffusion probabilistic models

- Training and sample generation in DDPM:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Data  Noise

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boxed{\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)} + \sigma_t\mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

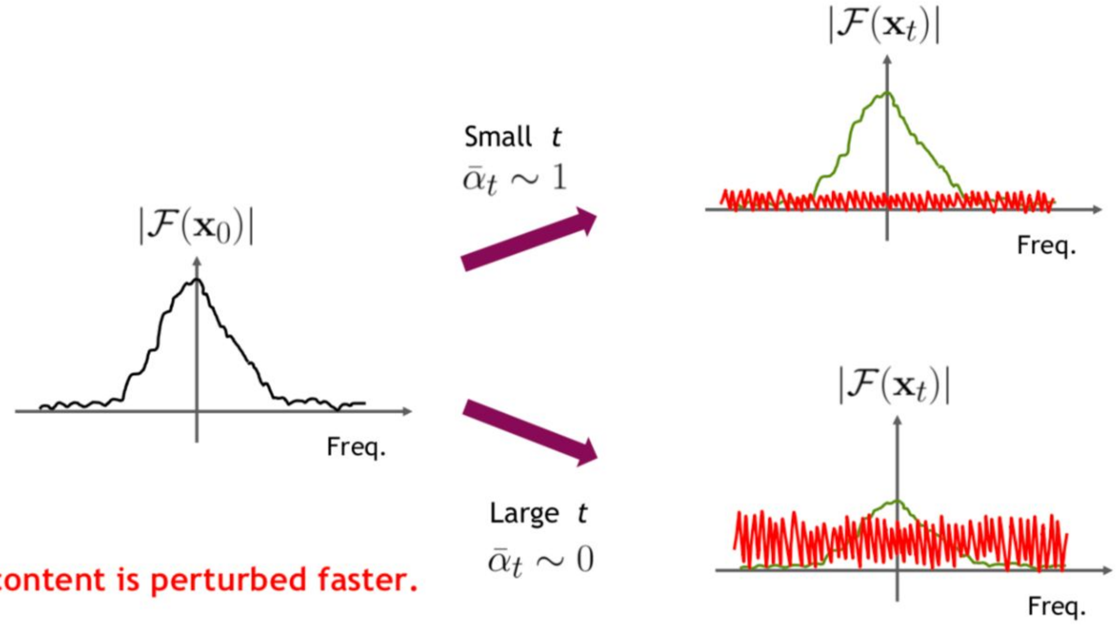
Denoising diffusion probabilistic models

- Training and sample generation in DDPM:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon$$

$\Downarrow$ Fourier Transform

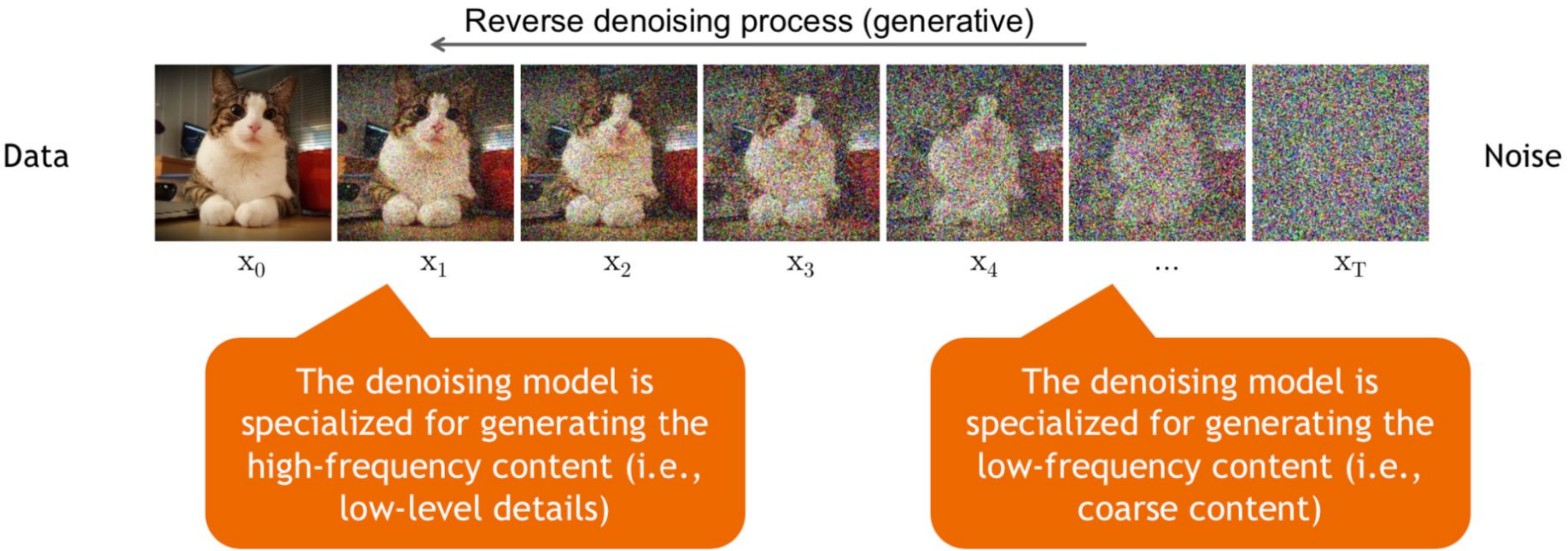
$$\mathcal{F}(\mathbf{x}_t) = \sqrt{\bar{\alpha}_t}\,\mathcal{F}(\mathbf{x}_0) + \sqrt{(1-\bar{\alpha}_t)}\,\mathcal{F}(\epsilon)$$



In the forward diffusion, the high frequency content is perturbed faster.

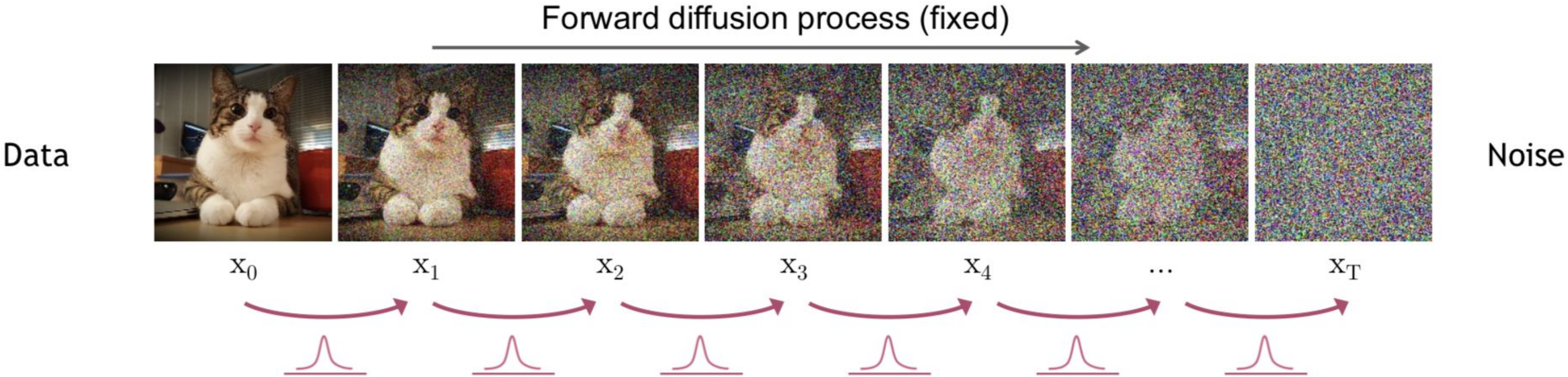
Denoising diffusion probabilistic models

- Training and sample generation in DDPM:



DDPM and score-based models

- Consider forward diffusion process in DDPM:

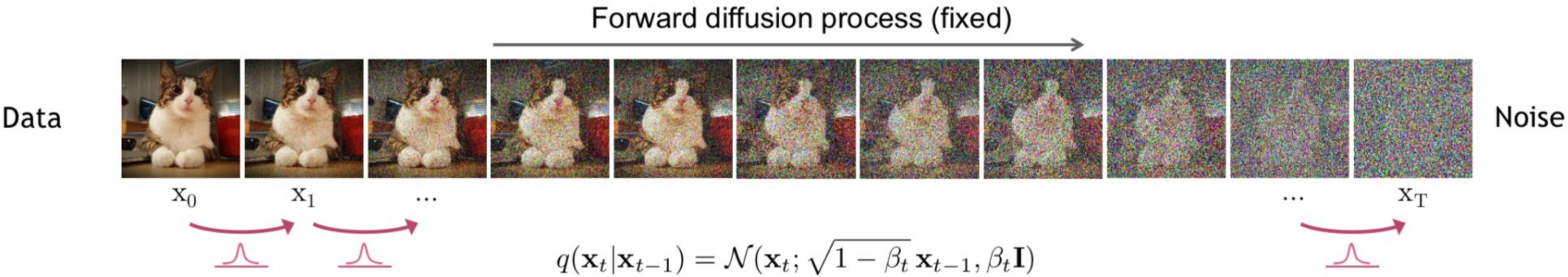


$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR, 2021

DDPM and score-based models

- Consider the limit of many small steps:



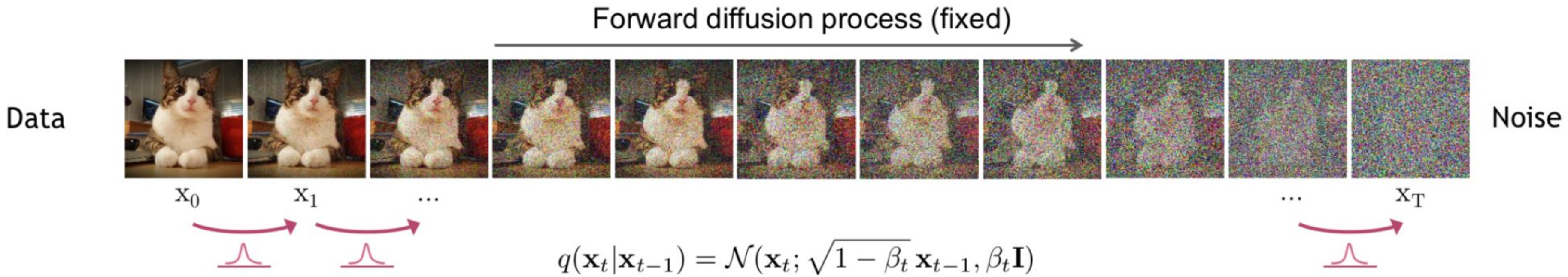
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

$$\Rightarrow \quad \mathbf{x}_t = \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$$

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR, 2021

DDPM and score-based models

- Consider the limit of many small steps:



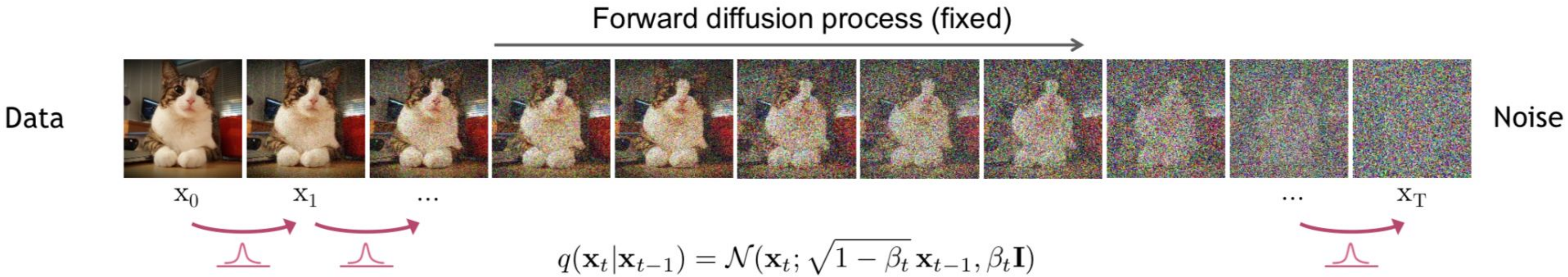
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$\begin{aligned}\mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\ &= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \qquad (\beta_t := \beta(t)\Delta t)\end{aligned}$$

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR, 2021

DDPM and score-based models

- Consider the limit of many small steps:



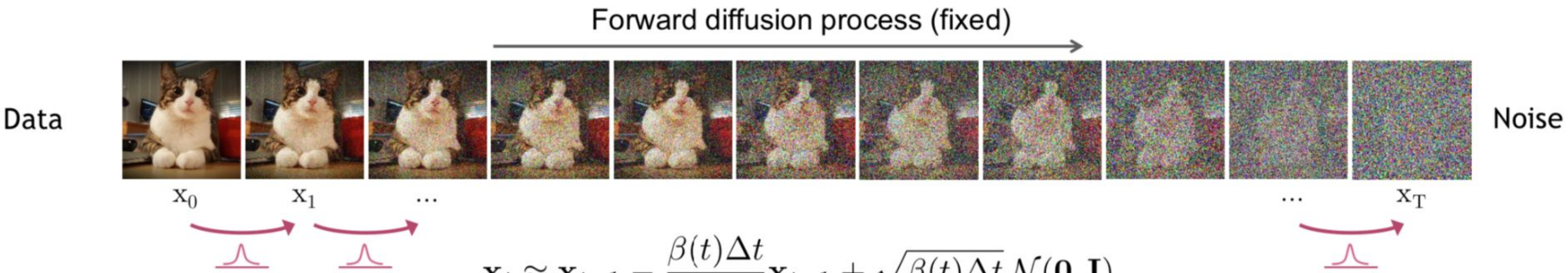
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$\begin{aligned} \mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\ &= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && (\beta_t := \beta(t)\Delta t) \\ \Rightarrow\quad &\approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && \text{(Taylor expansion)} \end{aligned}$$

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR, 2021

DDPM and score-based models

- Consider the limit of many small steps:



$$\mathbf{x}_t \approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$$

⬇

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Stochastic Differential Equation (SDE)
describing the diffusion in infinitesimal limit

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR, 2021

Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

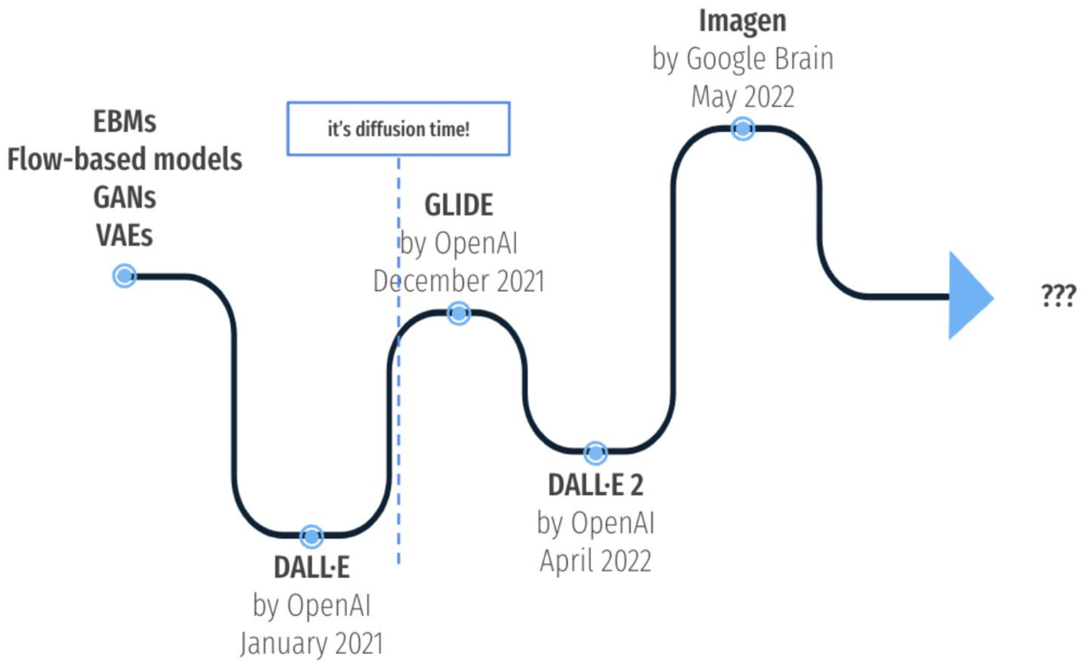
Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

**Some slides in this section are adapted from the post: [The recent rise of diffusion-based models](#)*

Image generation

- Timeline of image generation and text-to-image solutions



Text-guided diffusion with GLIDE

- Model architecture components:
  - A U-Net based model responsible for the visual part of the diffusion learning
  - A transformer-based model responsible for creating a text embedding from a snippet of text
  - An upsampling diffusion model is used for enhancing output image resolution

"a hedgehog using a calculator"

"a corgi wearing a red bowtie and a purple party hat"

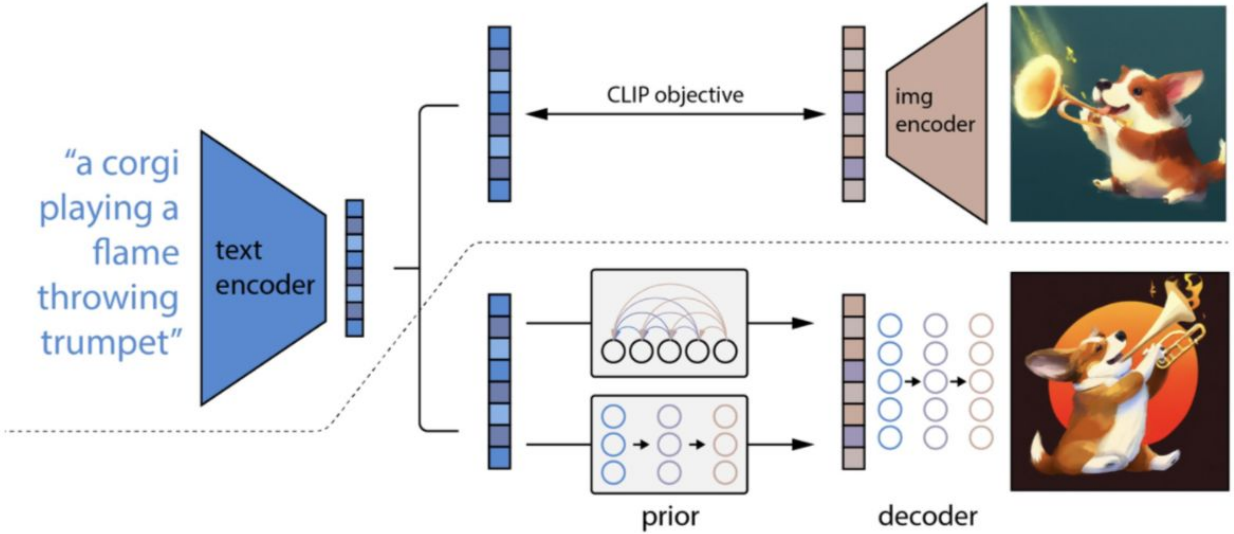
"robots meditating in a vipassana retreat"

"a fall landscape with a small cottage next to a lake"

Nichol et al., GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models, 2021

DALL · E 2

- Model architecture components:
  - Prior: produces CLIP image embeddings conditioned on the caption
  - Decoder: produces images conditioned on CLIP image embeddings and text



Ramesh et al., Hierarchical Text-Conditional Image Generation with CLIP Latents, arXiv 2022

DALL·E 2

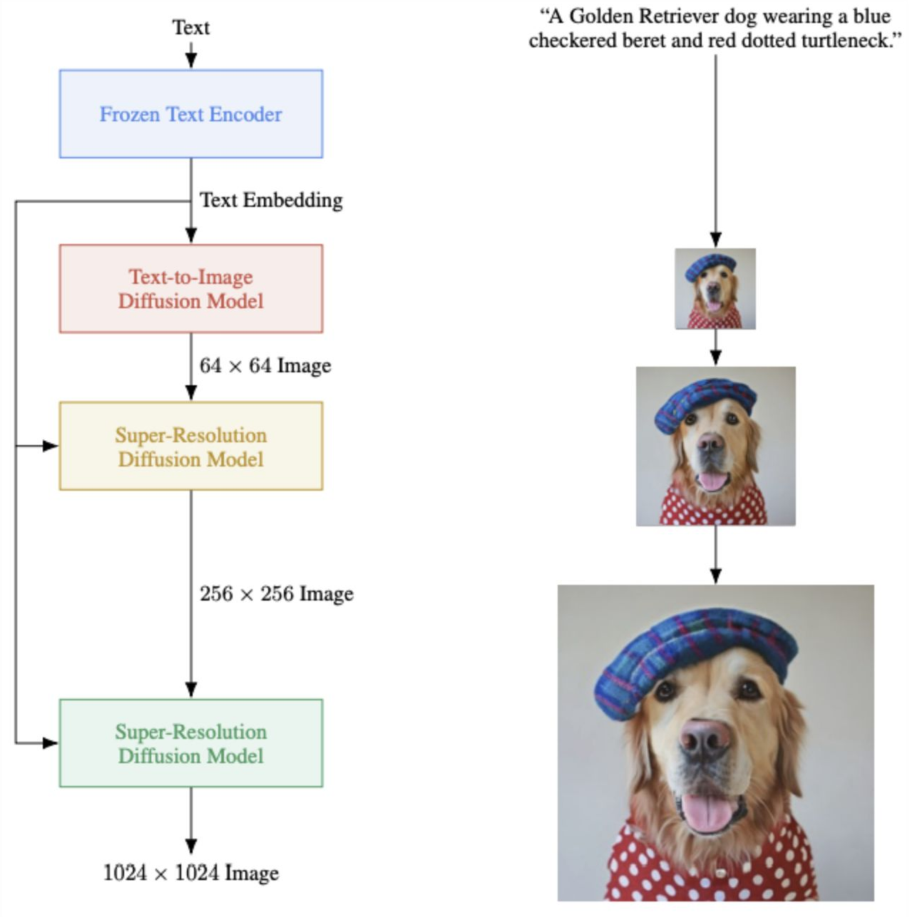
- Generation examples

"People walking on a beach during sunrise, a reflection of the sun on the water, realistic"

"Flamingos standing on water, red sunset, pink-red water reflection, photo-realistic, 4k"

Imagen

- Model architecture components:
  - Use a frozen large pretrained language models as text encoder (T5-XXL)
  - More efficient neural network as backbone of cascaded super-resolution diffusion models (Efficient U-net)
  - More technical details in the paper to enhance image fidelity (conditioning augmentation, etc.)



Saharia et al., Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, arXiv 2022

Imagen

- Generation examples

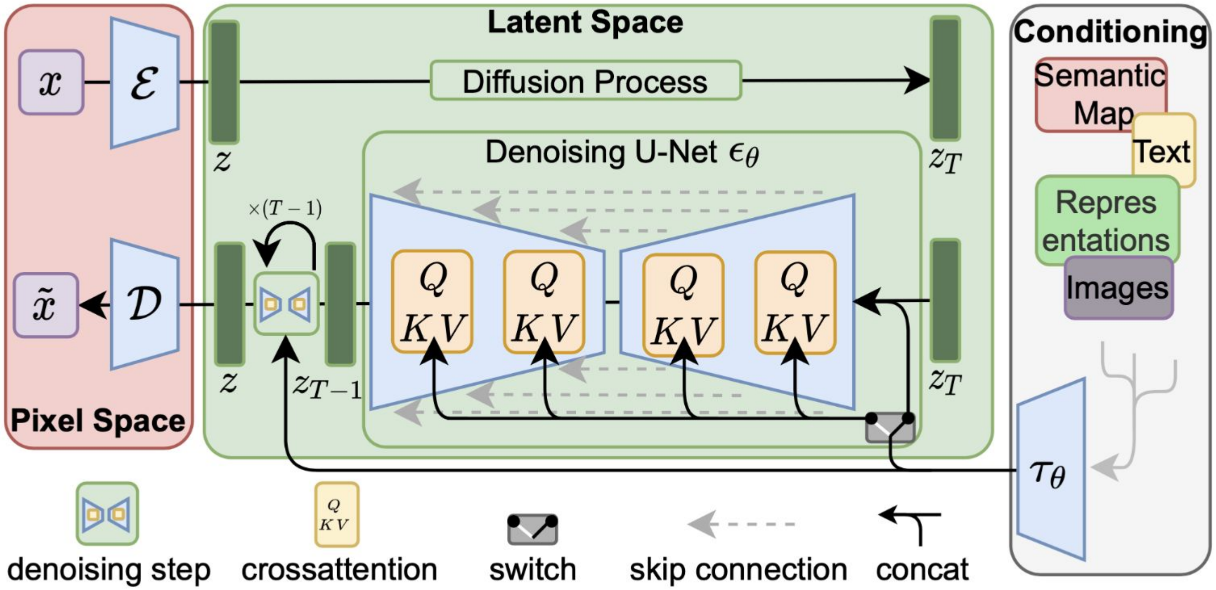
"A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat."

"A blue jay standing on a large basket of rainbow macarons"

"A brain riding a rocketship heading towards the moon"

Latent diffusion model (LDM)

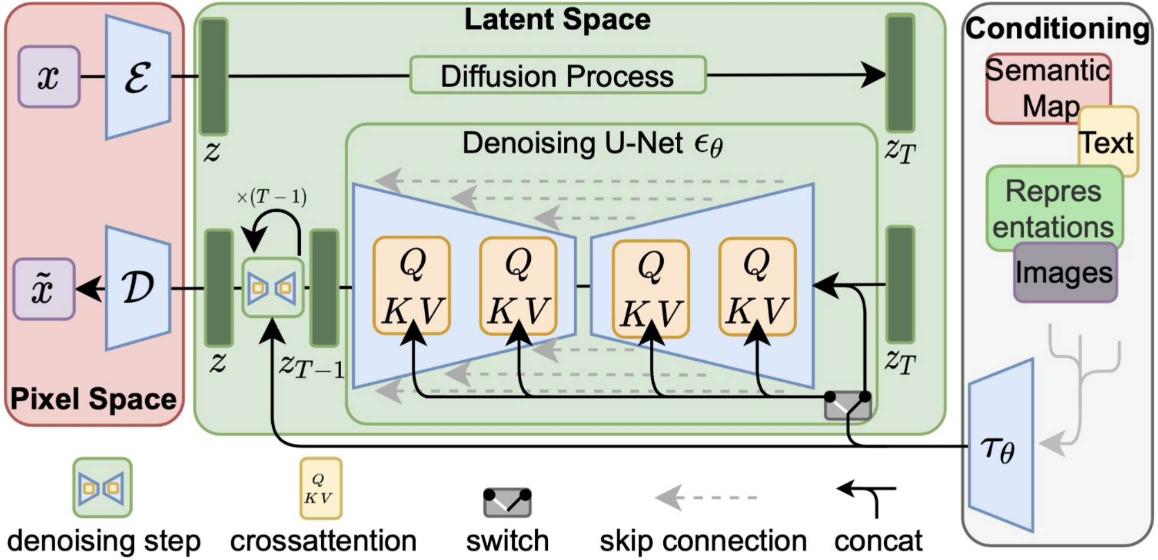
- Runs diffusion process in the latent space instead of pixel space
- Make training cost lower and inference speed faster



Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models, CVPR 2022

Latent diffusion model (LDM)

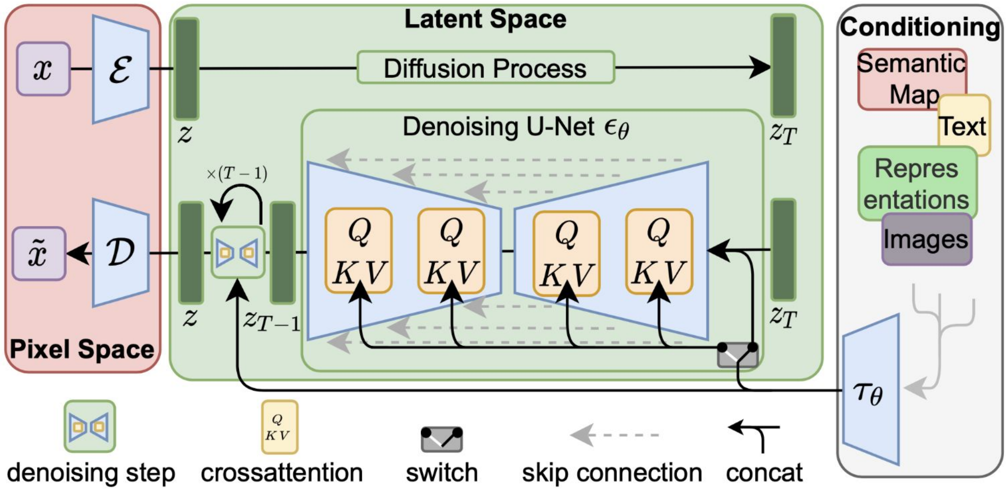
- Diffusion and denoising process:
  - Happen on the latent vector z
  - Denoising model is a time-conditioned U-Net on the latent vector z



Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models, CVPR 2022

Latent diffusion model (LDM)

- Cross-attention mechanism:
  - Fuse flexible conditioning information for image generation
  - Domain-specific encoder to project the conditioning input to an intermediate representation



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

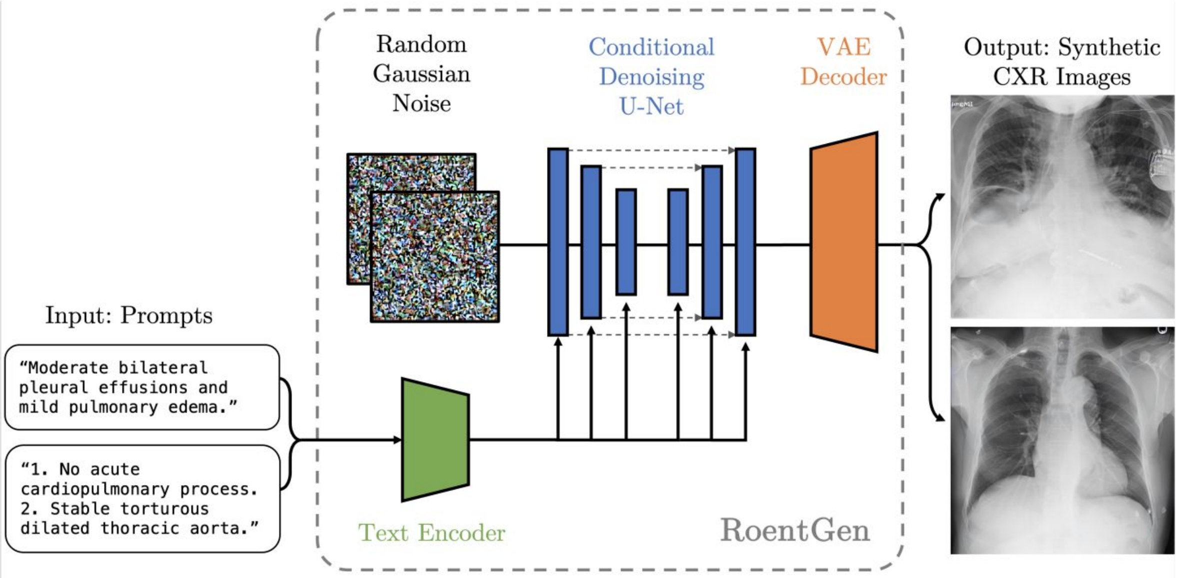
where $\mathbf{Q} = \mathbf{W}_Q^{(i)}\cdot\varphi_i(\mathbf{z}_i),\ \mathbf{K} = \mathbf{W}_K^{(i)}\cdot\tau_\theta(y),\ \mathbf{V} = \mathbf{W}_V^{(i)}\cdot\tau_\theta(y)$

and $\mathbf{W}_Q^{(i)} \in \mathbb{R}^{d\times d_\epsilon^i},\ \mathbf{W}_K^{(i)},\mathbf{W}_V^{(i)} \in \mathbb{R}^{d\times d_\tau},\ \varphi_i(\mathbf{z}_i) \in \mathbb{R}^{N\times d_\epsilon^i},\ \tau_\theta(y) \in \mathbb{R}^{M\times d_\tau}$

Rombach et al., High-Resolution Image Synthesis with Latent Diffusion Models, CVPR 2022

Medical image generation

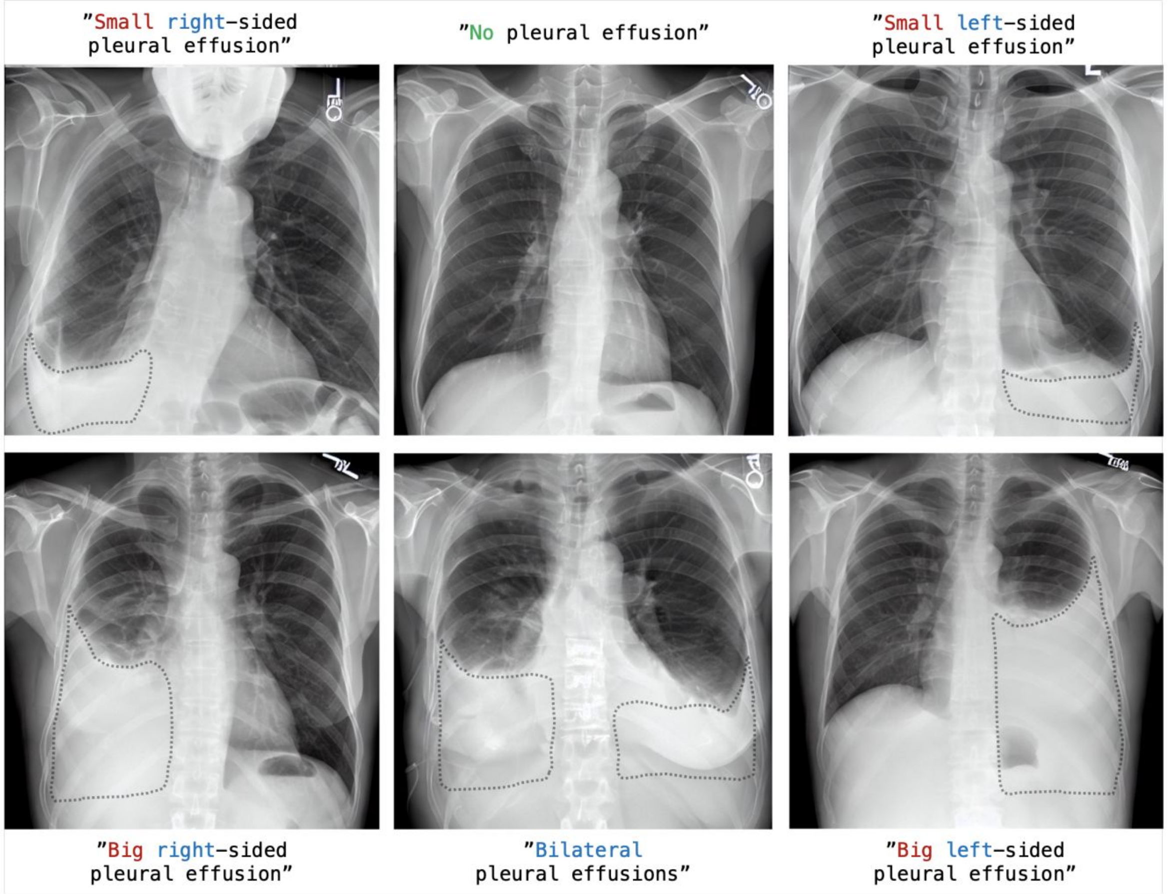
- RoentGen: text-to-image synthesis of chest x-ray images
  - A medical domain-adapted latent diffusion model based on the Stable Diffusion pipeline
  - Fine-tuned or retrain conditional U-Net with free-text prompts through a medical text encoder



Chambon et al., RoentGen: Vision-Language Foundation Model for Chest X-ray Generation, arXiv 2022

Medical image generation

- Generation examples



Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

Controllable generation for inverse problem solving

- Diffusion generative models are particularly suitable for solving inverse problems
  - Inverse problem:
    - Given measurements y
    - Known forward process of generating y from x: p(y | x)
    - To compute posterior distribution: p(x | y)
  - From Bayes' rule:

$$p(\mathbf{x} \mid \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})\mathrm{d}\mathbf{x}.$$

  - Compute the posterior score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x})$$

Controllable generation for inverse problem solving

- Diffusion generative models are particularly suitable for solving inverse problems
  - Inverse problem:
    - Given measurements y
    - Known forward process of generating y from x: p(y | x)
    - To compute posterior distribution: p(x | y)
  - From Bayes' rule:

$$p(\mathbf{x} \mid \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})\mathrm{d}\mathbf{x}.$$

  - Compute the posterior score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y}) = \boxed{\nabla_{\mathbf{x}} \log p(\mathbf{x})} + \nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x})$$

Score function of unconditional data distribution !

Controllable generation for inverse problem solving

- Diffusion generative models are particularly suitable for solving inverse problems
  - Inverse problem:
    - Given measurements y
    - Known forward process of generating y from x: p(y | x)
    - To compute posterior distribution: p(x | y)
  - From Bayes' rule:

$$p(\mathbf{x} \mid \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x}) / \int p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})\mathrm{d}\mathbf{x}.$$

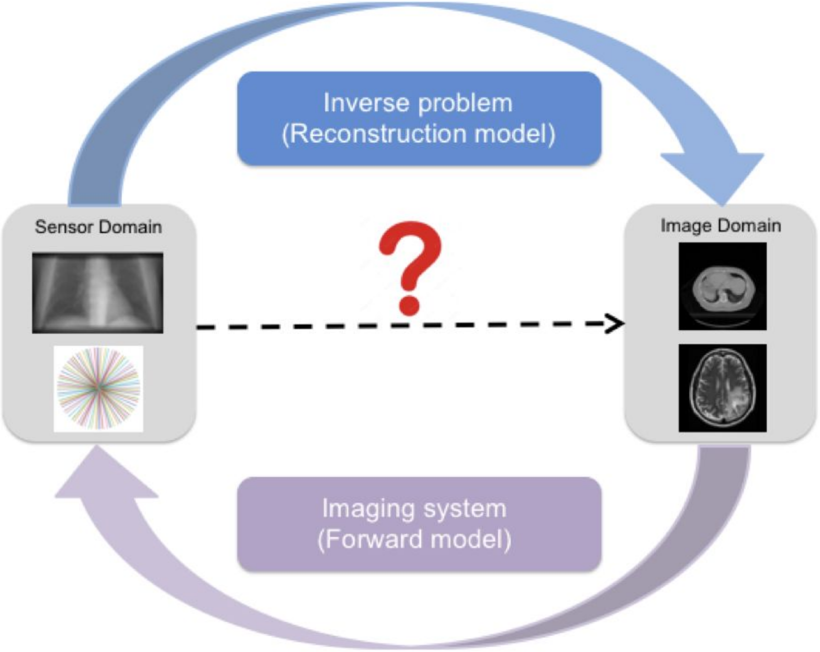
  - Compute the posterior score function:

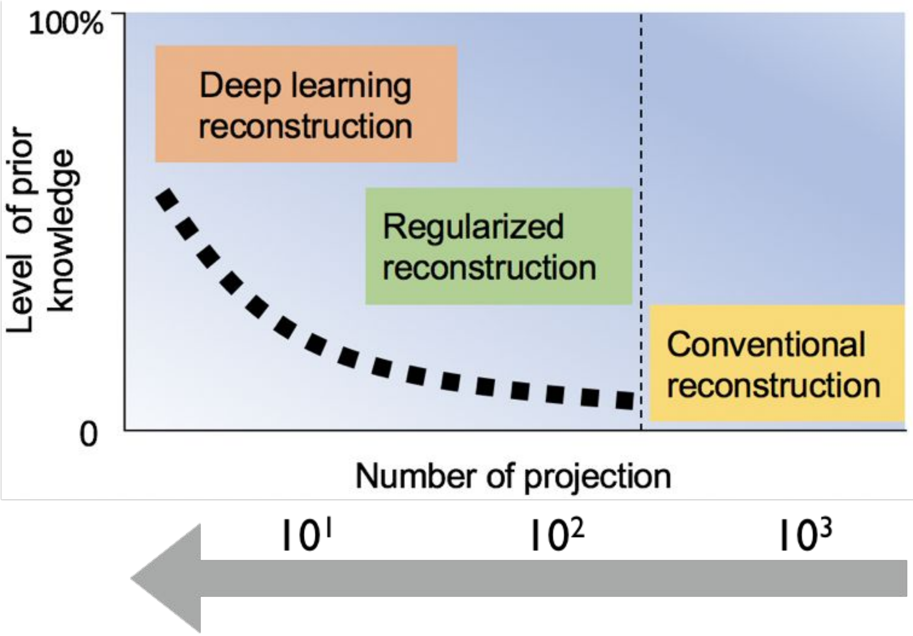
$$\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mathbf{y}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \boxed{\nabla_{\mathbf{x}} \log p(\mathbf{y} \mid \mathbf{x})}$$

Known forward process

Inverse problems in biomedical imaging

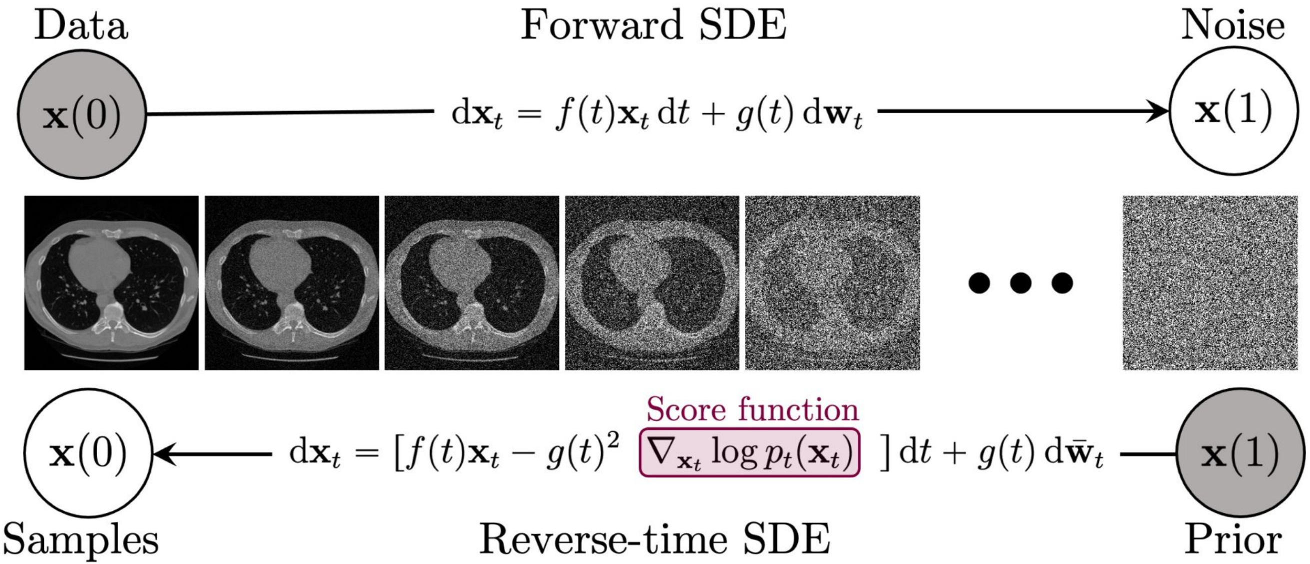
- Sparse-sampling image reconstruction
  - Leverage prior knowledge for solving ill-posed inverse problem





Inverse problems in biomedical imaging

- Diffusion generative models provide population priors by modeling data distribution
  - Perturbation process: forward SDE
  - Sampling process: reverse-time SDE

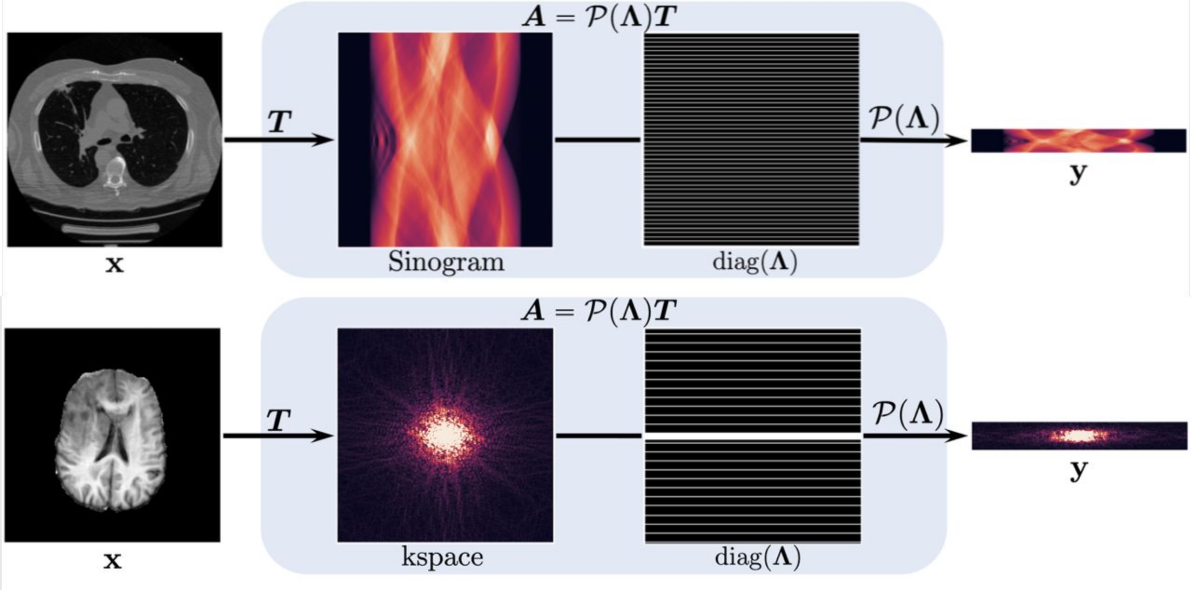


Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Measurement process for sparse-sampling biomedical imaging

- Forward model:
  - Physics-based transformation
  - Sparse sampling mask of observed measurements

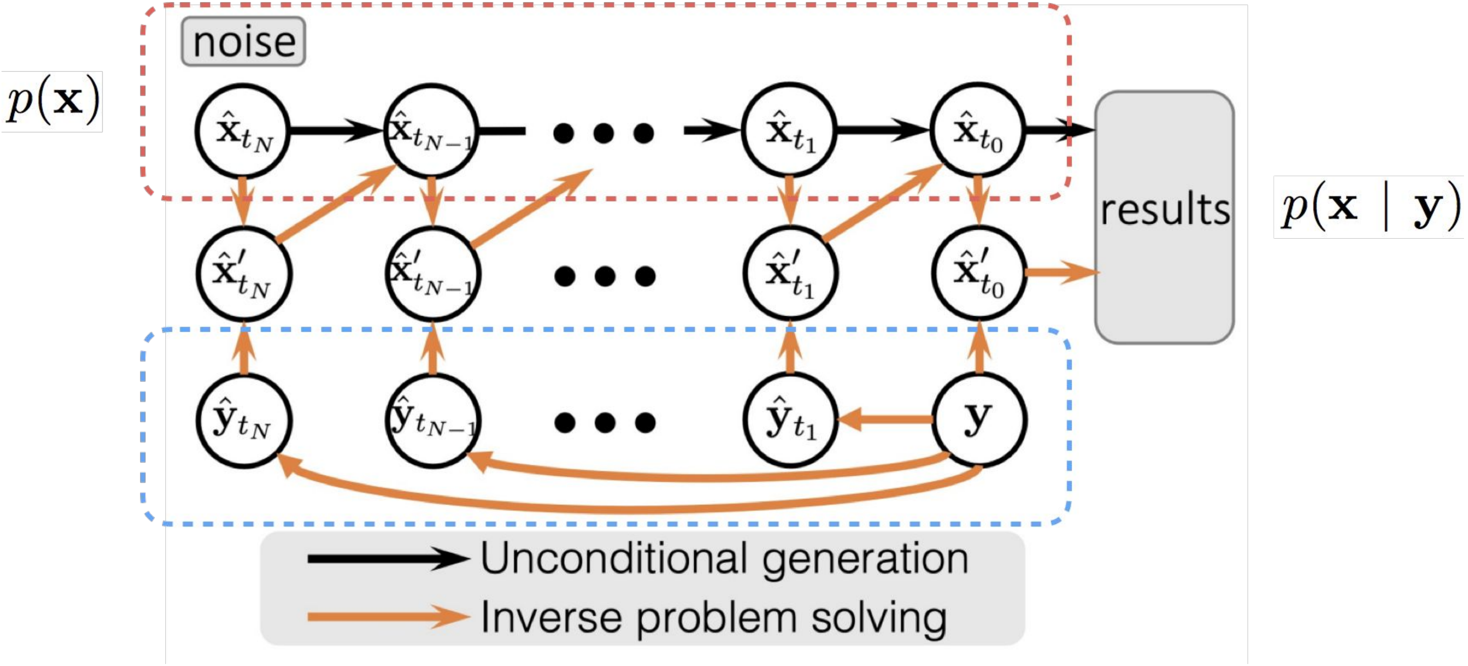
x: image
y: observed measurements
A: forward model
T: transformation
Λ: sampling mask
P: dimension reduction operator



Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Conditional sampling approach for inverse problem solving

- Generate image consistent with population prior and observed measurements

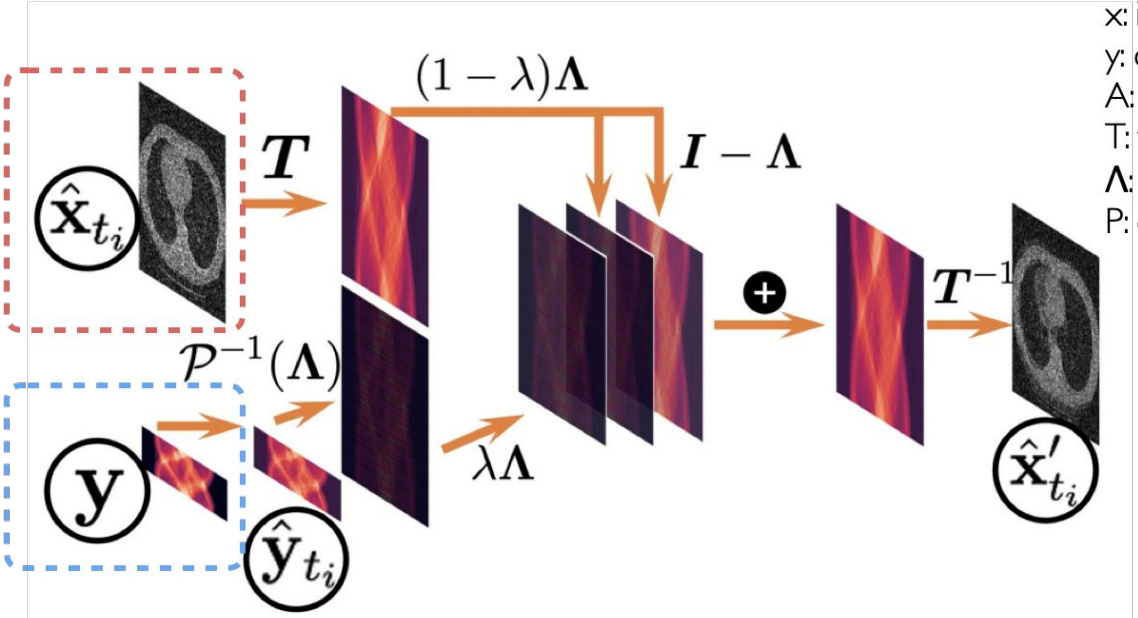


Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Optimization objective

- Generate image consistent with population prior and observed measurements

$$\hat{\mathbf{x}}'_{t_i} = \arg\min_{\boldsymbol{z}\in\mathbb{R}^n}\{(1-\lambda)\|\boldsymbol{z}-\hat{\mathbf{x}}_{t_i}\|_{\boldsymbol{T}}^2 + \min_{\boldsymbol{u}\in\mathbb{R}^n}\lambda\|\boldsymbol{z}-\boldsymbol{u}\|_{\boldsymbol{T}}^2\} \quad s.t. \quad \boldsymbol{A}\boldsymbol{u} = \hat{\mathbf{y}}_{t_i}$$

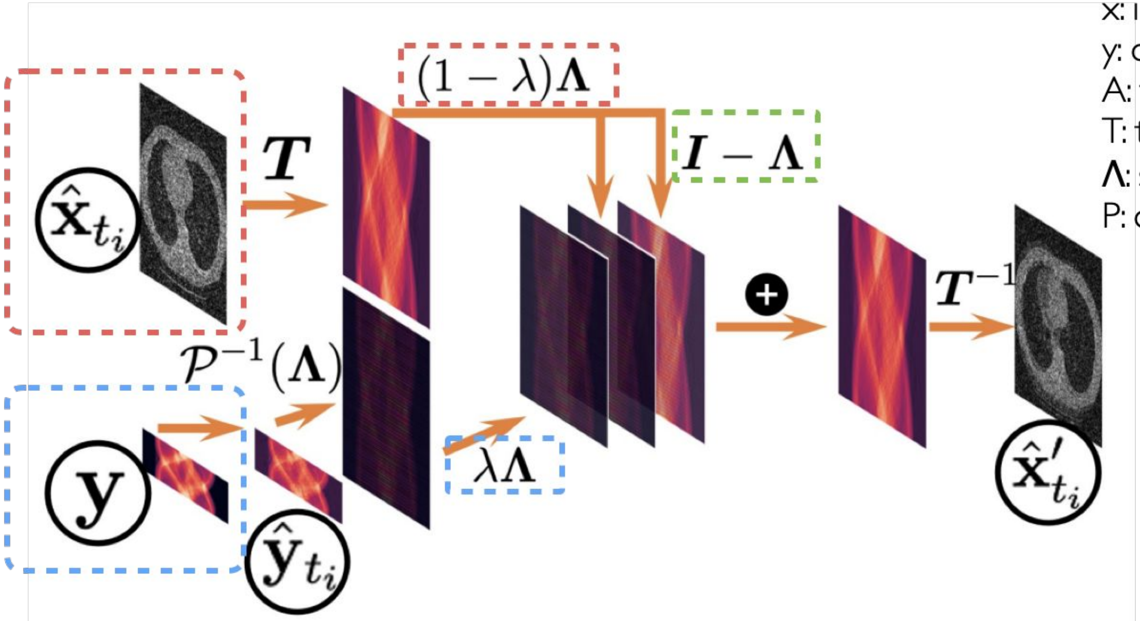


x: image
y: observed measurements
A: forward model
T: transformation
Λ: sampling mask
P: dimension reduction operator

Solution

- Generate image consistent with population prior and observed measurements

$$\hat{\mathbf{x}}'_{t_i} = \boldsymbol{T}^{-1}[(\boldsymbol{I} - \boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_{t_i} + (1-\lambda)\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_{t_i} + \lambda\boldsymbol{\Lambda}\mathcal{P}^{-1}(\boldsymbol{\Lambda})\hat{\mathbf{y}}_{t_i}]$$



x: image
y: observed measurements
A: forward model
T: transformation
Λ: sampling mask
P: dimension reduction operator

Conditional sampling approach for inverse problem solving

- Generate image consistent with population prior and observed measurements

Algorithm 1 Unconditional sampling

Require: N

$\hat{\mathbf{x}}_1 \sim \pi(\mathbf{x}), \Delta t \leftarrow \frac{1}{N}$
for $i = N - 1$ **to** 0 **do**
  $t \leftarrow \frac{i+1}{N}$
  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_t - f(t)\hat{\mathbf{x}}_t \Delta t$
  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_{t-\Delta t} + g(t)^2 \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_t, t)\Delta t$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_{t-\Delta t} + g(t)\sqrt{\Delta t}\, \mathbf{z}$
return $\hat{\mathbf{x}}_0$

Algorithm 2 Inverse problem solving

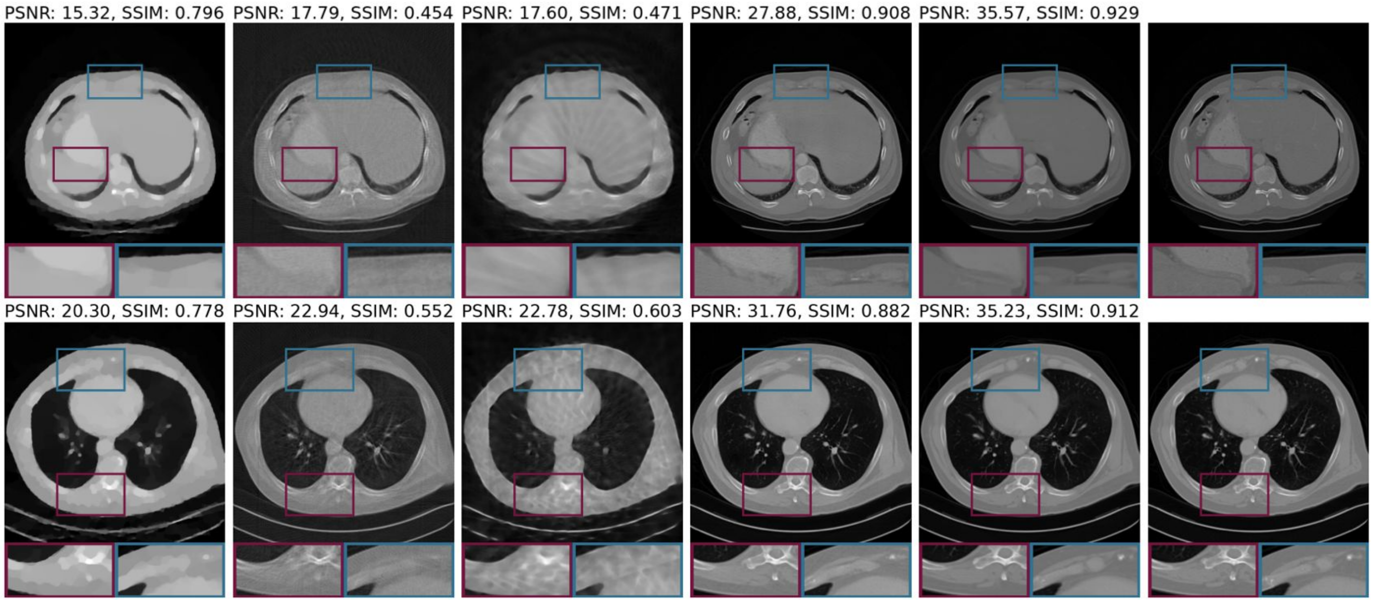
Require: $N, \mathbf{y}, \lambda$

$\hat{\mathbf{x}}_1 \sim \pi(\mathbf{x}), \Delta t \leftarrow \frac{1}{N}$
for $i = N - 1$ **to** 0 **do**
  $t \leftarrow \frac{i+1}{N}$
  $\hat{\mathbf{y}}_t \sim p_{0t}(\mathbf{y}_t \mid \mathbf{y})$
  $\hat{\mathbf{x}}_t \leftarrow \boldsymbol{T}^{-1}[\lambda \boldsymbol{\Lambda} \mathcal{P}^{-1}(\boldsymbol{\Lambda})\hat{\mathbf{y}}_t + (1-\lambda)\boldsymbol{\Lambda}\boldsymbol{T}\hat{\mathbf{x}}_t + (\boldsymbol{I} - \boldsymbol{\Lambda})\boldsymbol{T}\hat{\mathbf{x}}_t]$
  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_t - f(t)\hat{\mathbf{x}}_t \Delta t$
  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_{t-\Delta t} + g(t)^2 \boldsymbol{s}_{\boldsymbol{\theta}*}(\hat{\mathbf{x}}_t, t)\Delta t$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\hat{\mathbf{x}}_{t-\Delta t} \leftarrow \hat{\mathbf{x}}_{t-\Delta t} + g(t)\sqrt{\Delta t}\, \mathbf{z}$
return $\hat{\mathbf{x}}_0$

Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Results for medical image reconstruction

- Data-efficiency: unsupervised learning technique
    - No assumption of measurements sampling during training
    - Comparable or better performance to supervised learning methods



(a) FISTA-TV (b) cGAN (c) Neumann (d) SIN-4c-PRN (e) Ours (f) Ground truth

Results for medical image reconstruction

- Generalization: flexibly adapt to different sampling processes at test time
  - Different number of projections in CT reconstruction

Method	Projections	LIDC 320 × 320		LDCT 512 × 512	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑
FBP	23	10.18±1.38	0.230±0.072	10.11±1.19	0.302±0.078
FISTA-TV	23	20.08±4.89	0.799±0.061	21.88±4.42	0.850±0.067
cGAN	23	19.83±3.07	0.479±0.103	19.90±2.52	0.545±0.065
Neumann	23	17.18±3.79	0.454±0.128	18.83±3.29	0.525±0.073
SIN-4c-PRN	23	30.48±3.99	0.895±0.047	34.82±3.55	0.877±0.116
Ours	10	29.52±2.63	0.823±0.061	28.96±4.41	0.849±0.086
	20	34.40±2.66	0.895±0.048	36.80±4.50	0.936±0.058
	23	**35.24**±2.71	**0.905**±0.046	**37.41**±4.62	**0.941**±0.057

Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Results for medical image reconstruction

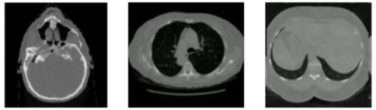
- Generalization: flexibly adapt to different sampling processes at test time
  - Different acceleration factors in MRI reconstruction

Method	24× Acceleration		8× Acceleration		4× Acceleration	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Cascade DenseNet	23.39±2.17	0.765±0.042	28.35±2.30	0.845±0.038	30.97±2.33	0.902±0.028
DuDoRNet	18.46±3.05	0.662±0.093	**37.88±3.03**	**0.985±0.007**	30.53±4.13	0.891±0.071
Score SDE	27.83±2.73	0.849±0.038	35.04±2.11	0.943±0.016	37.55±2.08	0.960±0.013
Langevin	28.80±3.21	0.873±0.039	36.44±2.28	0.952±0.016	38.76±2.32	0.966±0.012
Ours	**29.42±3.03**	**0.880±0.035**	37.63±2.70	0.958±0.015	**39.91±2.67**	**0.965±0.013**

Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Results for medical image reconstruction

- Generalization:
  - Capture population prior of multi-anatomic site images

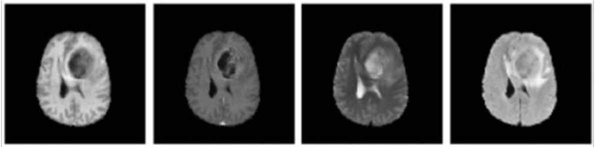


Method	Projections	LIDC 320 × 320		LDCT 512 × 512	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑
FBP	23	10.18±1.38	0.230±0.072	10.11±1.19	0.302±0.078
FISTA-TV	23	20.08±4.89	0.799±0.061	21.88±4.42	0.850±0.067
cGAN	23	19.83±3.07	0.479±0.103	19.90±2.52	0.545±0.065
Neumann	23	17.18±3.79	0.454±0.128	18.83±3.29	0.525±0.073
SIN-4c-PRN	23	30.48±3.99	0.895±0.047	34.82±3.55	0.877±0.116
Ours	10	29.52±2.63	0.823±0.061	28.96±4.41	0.849±0.086
	20	34.40±2.66	0.895±0.048	36.80±4.50	0.936±0.058
	23	**35.24**±2.71	**0.905**±0.046	**37.41**±4.62	**0.941**±0.057

Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Results for medical image reconstruction

- Generalization:
  - Capture population prior of multi-modal images



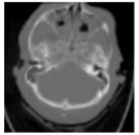
Method	24× Acceleration		8× Acceleration		4× Acceleration	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Cascade DenseNet	23.39±2.17	0.765±0.042	28.35±2.30	0.845±0.038	30.97±2.33	0.902±0.028
DuDoRNet	18.46±3.05	0.662±0.093	**37.88±3.03**	**0.985±0.007**	30.53±4.13	0.891±0.071
Score SDE	27.83±2.73	0.849±0.038	35.04±2.11	0.943±0.016	37.55±2.08	0.960±0.013
Langevin	28.80±3.21	0.873±0.039	36.44±2.28	0.952±0.016	38.76±2.32	0.966±0.012
Ours	**29.42±3.03**	**0.880±0.035**	37.63±2.70	0.958±0.015	**39.91±2.67**	**0.965±0.013**

Song et al., Solving inverse problems in medical imaging with score-based generative models, ICLR 2022.

Results for medical image reconstruction

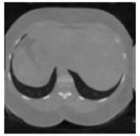
- Generalization:
  - A single training model captures population prior of multi-anatomic sites and multi-modal images

Multiple anatomic sites



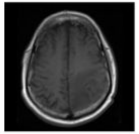
Head CT

Lung CT

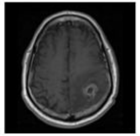


Abdominal CT

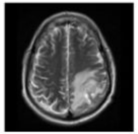
Multiple imaging modalities



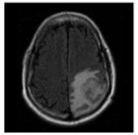
T1 MRI



T1-contrast MRI



T2 MRI



FLAIR MRI

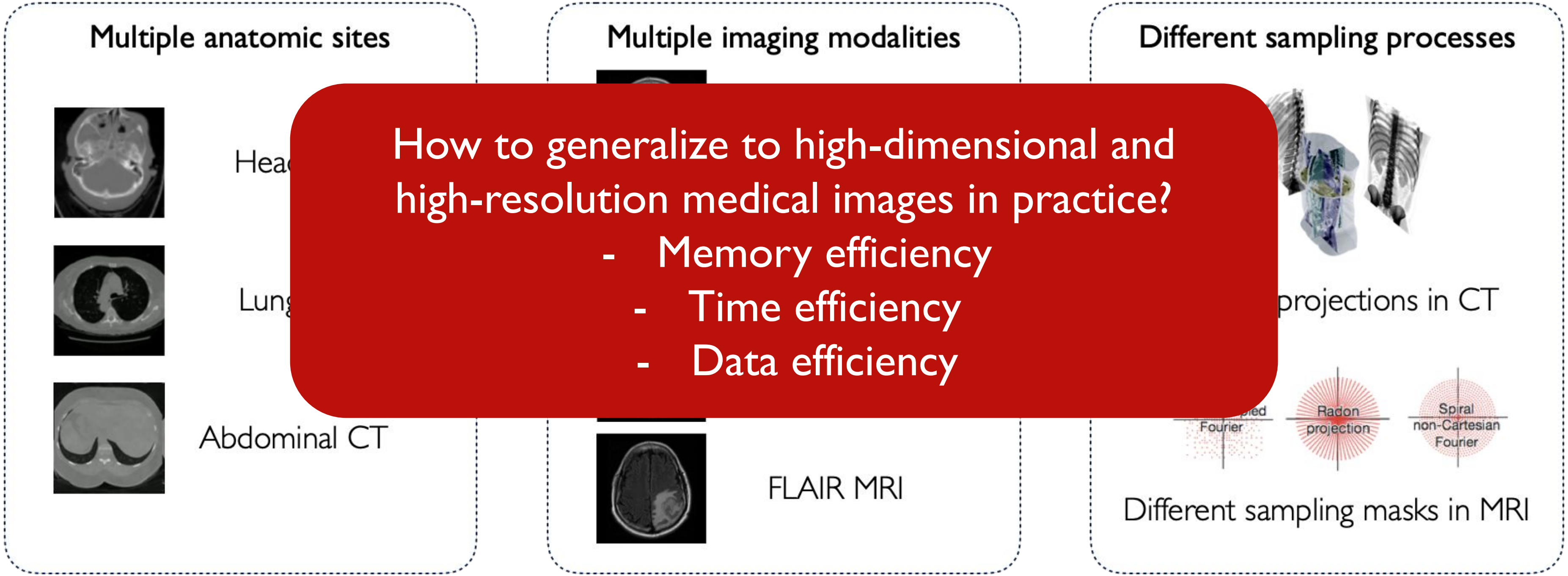
Different sampling processes

Different projections in CT

Different sampling masks in MRI

Results for medical image reconstruction

- Generalization:
  - A single training model captures population prior of multi-anatomic sites and multi-modal images

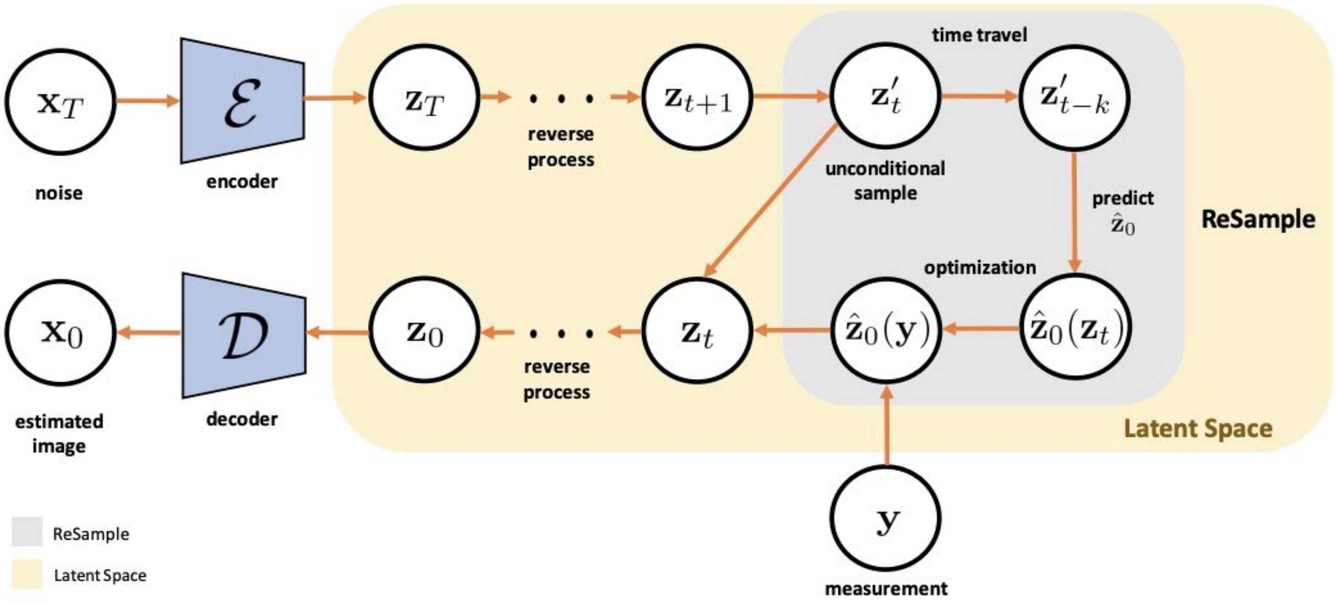


How to generalize to high-dimensional and high-resolution medical images in practice?

- Memory efficiency
- Time efficiency
- Data efficiency

Latent diffusion model for solving inverse problems

- ReSample:
  - Solve general inverse problems with pre-trained latent diffusion models via hard data consistency



Song et al., Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency, arXiv 2023

Latent diffusion model for solving inverse problems

- ReSample:
  - Solve general inverse problems with pre-trained latent diffusion models via hard data consistency

Algorithm 1 `ReSample`

Require: Measurements $\mathbf{y}$, $\mathcal{A}(\cdot)$, Encoder $\mathcal{E}(\cdot)$, Decoder $\mathcal{D}(\cdot)$, Score function $s_\theta(\cdot, t)$, Parameters β_t, $\bar{\alpha}_t$, η, δ, λ, Time steps for data consistency $C = \{1, \ldots, T\}$, Time travel iterations k

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\mathbf{z}_T = \mathcal{E}(\mathbf{x}_T)$ ▷ Initial noise vector

for $t = T-1, \ldots, 0$ **do**

  $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

  $\hat{\mathbf{z}}_0 = \frac{1}{\sqrt{\bar{\alpha}_{t+1}}}(\mathbf{z}_{t+1} - \sqrt{1-\bar{\alpha}_{t+1}}\mathbf{s}_\theta(\mathbf{z}_{t+1}, t+1))$ ▷ Predict $\hat{\mathbf{z}}_0$

  $\mathbf{z}'_t = \sqrt{\bar{\alpha}_t}\hat{\mathbf{z}}_0 + \sqrt{1-\bar{\alpha}_t - \eta\delta^2}\mathbf{s}_\theta(\mathbf{z}_{t+1}, t+1) + \eta\delta\boldsymbol{\epsilon}_1$ ▷ Unconditional DDIM step

  if $t \in C$ **then**

    $\mathbf{z}'_{t-k} = \mathrm{TimeTravel}(\mathbf{z}'_t)$ ▷ Reverse sample k times

    $\hat{\mathbf{z}}_0(\mathbf{z}_t) = \frac{1}{\sqrt{\bar{\alpha}_{t-k}}}(\mathbf{z}'_{t-k} + (1-\bar{\alpha}_{t-k})s_\theta(\mathbf{z}'_{t-k}, t-k))$ ▷ Use Tweedie's formula

    $\boxed{\hat{\mathbf{z}}_0(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{z}} \frac{1}{2}\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{z}))\|_2^2 + \frac{\lambda}{2}\|\mathbf{z} - \hat{\mathbf{z}}_0(\mathbf{z}_t)\|_2^2}$

    $\mathbf{z}_t = \mathrm{StochasticResample}(\hat{\mathbf{z}}_0(\mathbf{y}), \mathbf{z}'_t)$ ▷ Map back to t

  else

    $\mathbf{z}_t = \mathbf{z}'_t$ ▷ Unconditional sampling if no data consistency

$\mathbf{x}_0 = \mathcal{D}(\mathbf{z}_0)$

Song et al., Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency, arXiv 2023

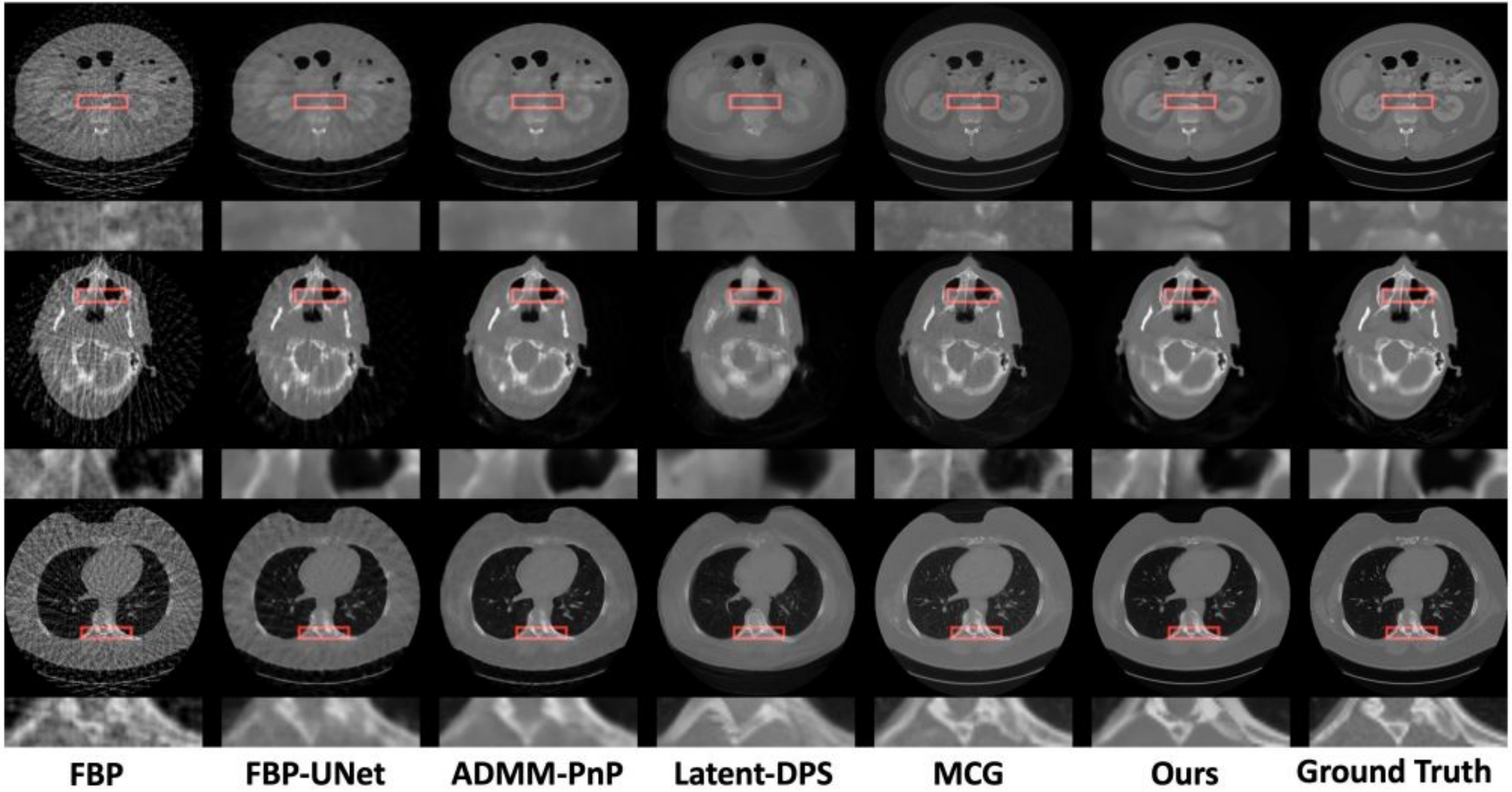
Latent diffusion model for solving inverse problems

- Results:
  - Achieve SOTA results for sparse-view CT reconstruction
  - Even outperform pixel-space diffusion models

Method	Abdominal		Head		Chest	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Latent-DPS	26.80	0.870	28.64	0.893	25.67	0.822
MCG [9]	29.41	0.857	28.28	0.795	27.92	0.842
DPS [6]	27.33	0.715	24.51	0.665	24.73	0.682
ADMM-PnP [2]	32.84	0.942	33.45	0.945	29.67	0.891
FBP	26.29	0.727	26.71	0.725	24.12	0.655
FBP-UNet [14]	32.77	0.937	31.95	0.917	29.78	0.885
ReSample (Ours)	**35.91**	**0.965**	**37.82**	**0.978**	**31.72**	**0.922**

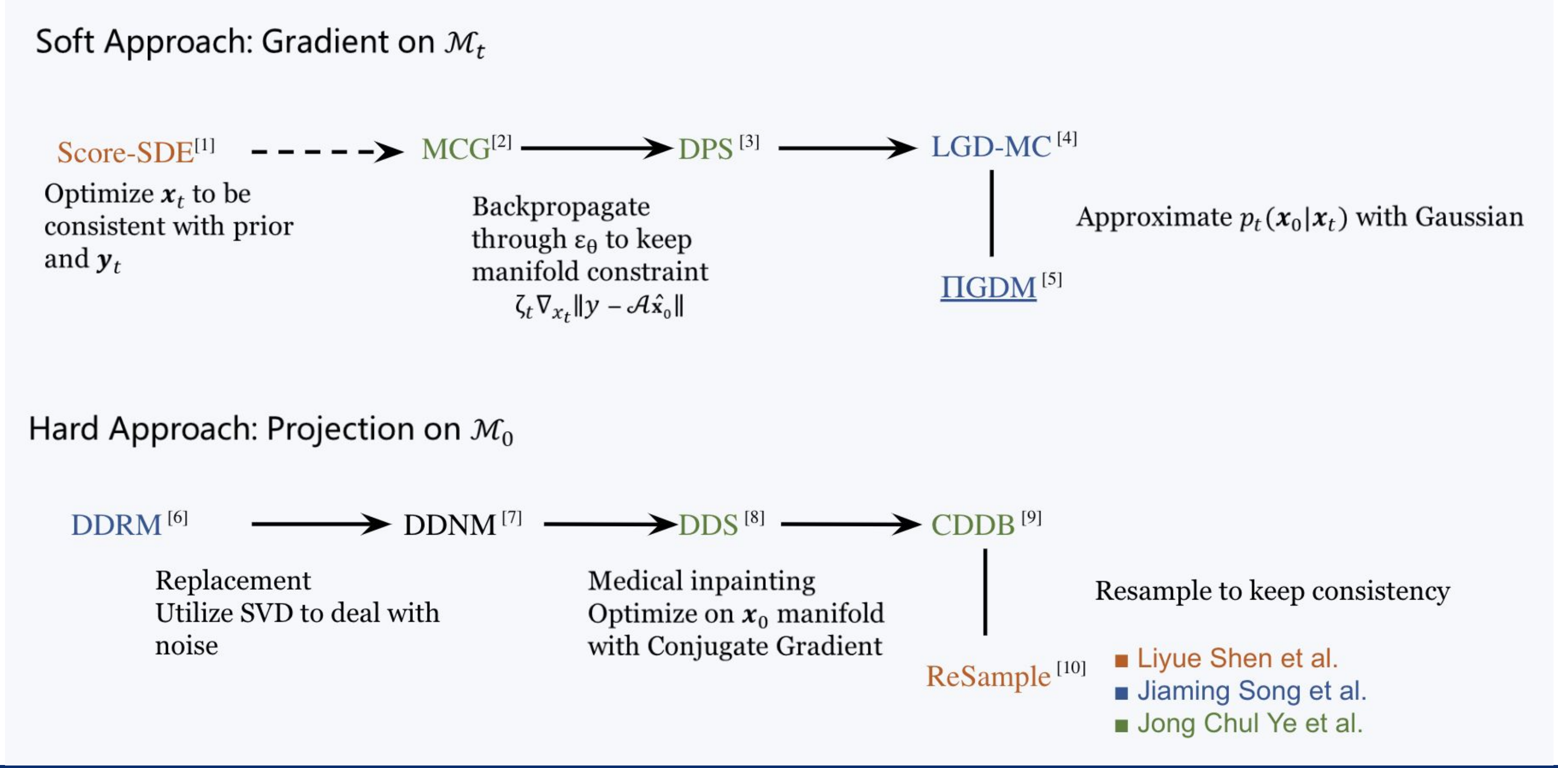
Song et al., Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency, arXiv 2023

Latent diffusion model for solving inverse problems



Diffusion model-based methods for solving inverse problems

- A lot of literatures…



Diffusion model-based methods for solving inverse problems

- A lot of literatures…
  - [1] Song, Yang, et al. "Solving Inverse Problems in Medical Imaging with Score-Based Generative Models." International Conference on Learning Representations.
  - [2] Chung, Hyungjin, et al. "Improving diffusion models for inverse problems using manifold constraints." Advances in Neural Information Processing Systems 35 (2022): 25683-25696.
  - [3] Chung, Hyungjin, et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems." The Eleventh International Conference on Learning Representations. 2022.
  - [4] Song, Jiaming, et al. "Loss-Guided Diffusion Models for Plug-and-Play Controllable Generation." (2023).
  - [5] Song, Jiaming, et al. "Pseudoinverse-guided diffusion models for inverse problems." International Conference on Learning Representations. 2022.
  - [6] Kawar, Bahjat, et al. "Denoising diffusion restoration models." Advances in Neural Information Processing Systems 35 (2022): 23593-23606.
  - [7] Wang, Yinhuai, Jiwen Yu, and Jian Zhang. "Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model." The Eleventh International Conference on Learning Representations. 2022.
  - [8] Chung, Hyungjin, Suhyeon Lee, and Jong Chul Ye. "Fast Diffusion Sampler for Inverse Problems by Geometric Decomposition." arXiv preprint arXiv:2303.05754 (2023).
  - [9] Chung, Hyungjin, Jeongsol Kim, and Jong Chul Ye. "Direct Diffusion Bridge using Data Consistency for Inverse Problems." arXiv preprint arXiv:2305.19809 (2023).
  - [10] Song, Bowen, et al. "Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency." arXiv preprint arXiv:2307.08123 (2023).
  - ……

Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

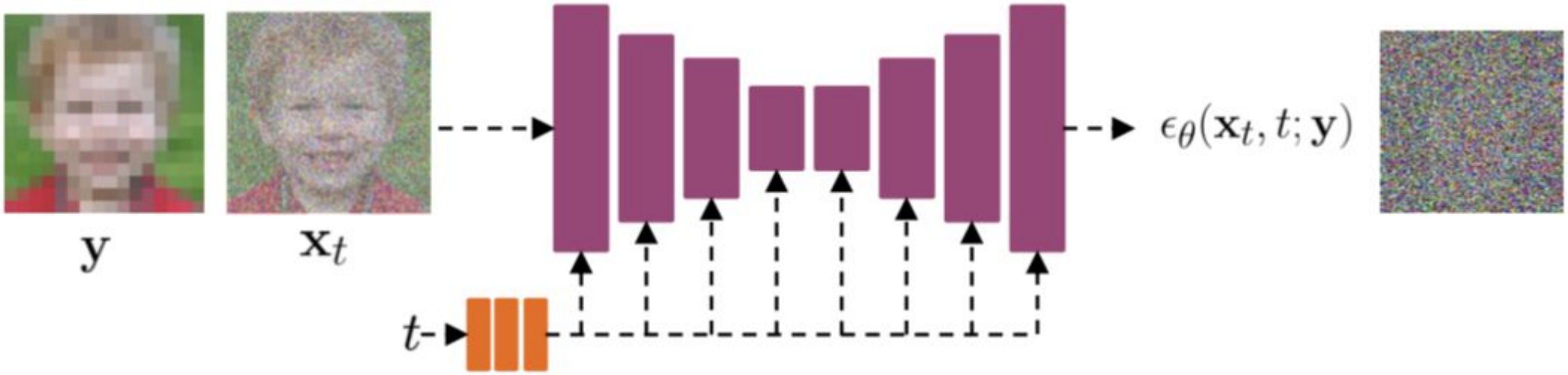
Today’s agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

**Some slides in this section are adapted from: [CVPR 2022 Tutorial: Denoising Diffusion-based Generative Modeling: Foundations and Applications]*

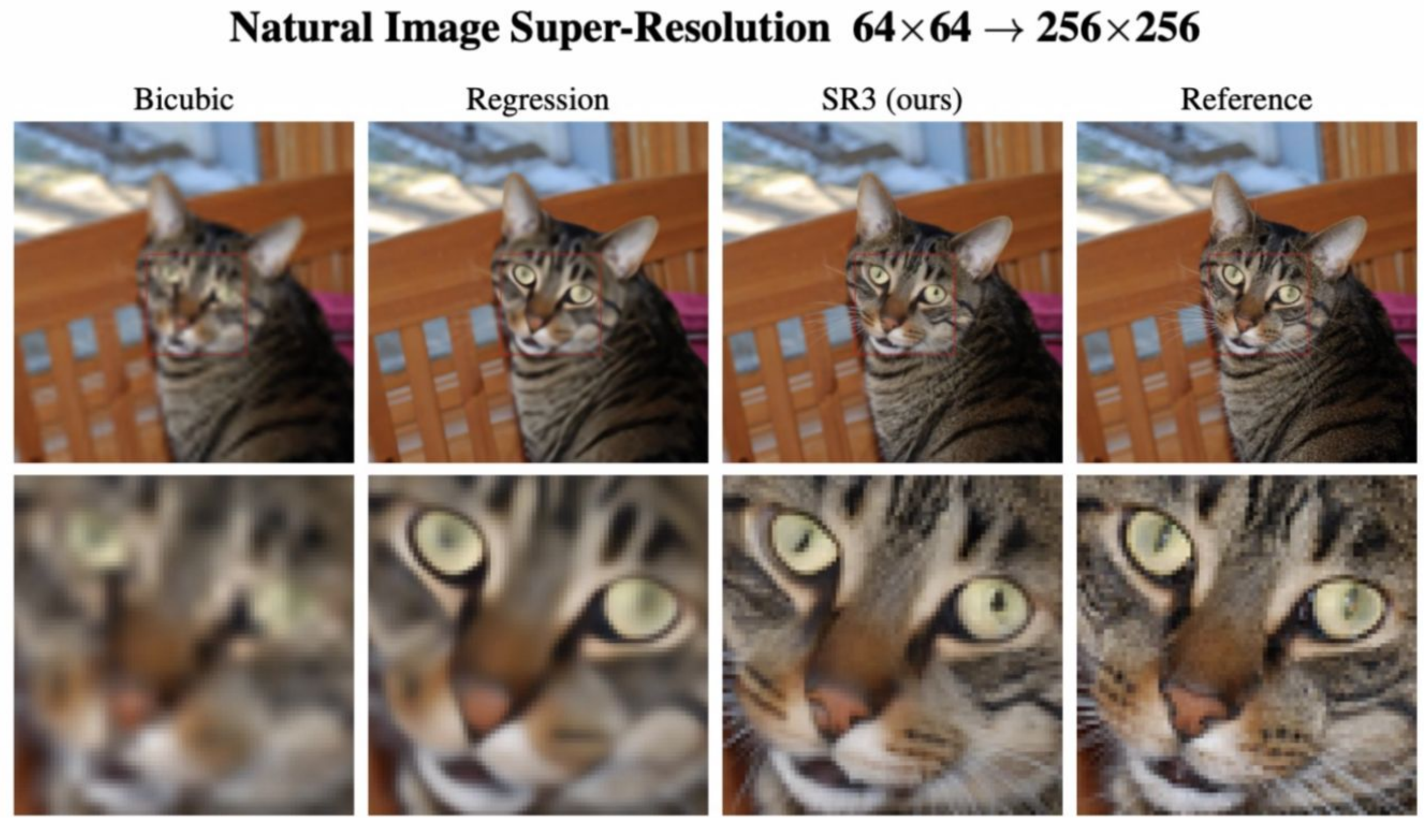
Super-resolution

- Image super-resolution can be considered as training p(x | y) where y is a low-resolution image and x is the corresponding high-resolution image
- The conditional score is simply a U-Net with x_t and y (low-resolution image) concatenated



Saharia et al., Image Super-Resolution via Iterative Refinement, 2021

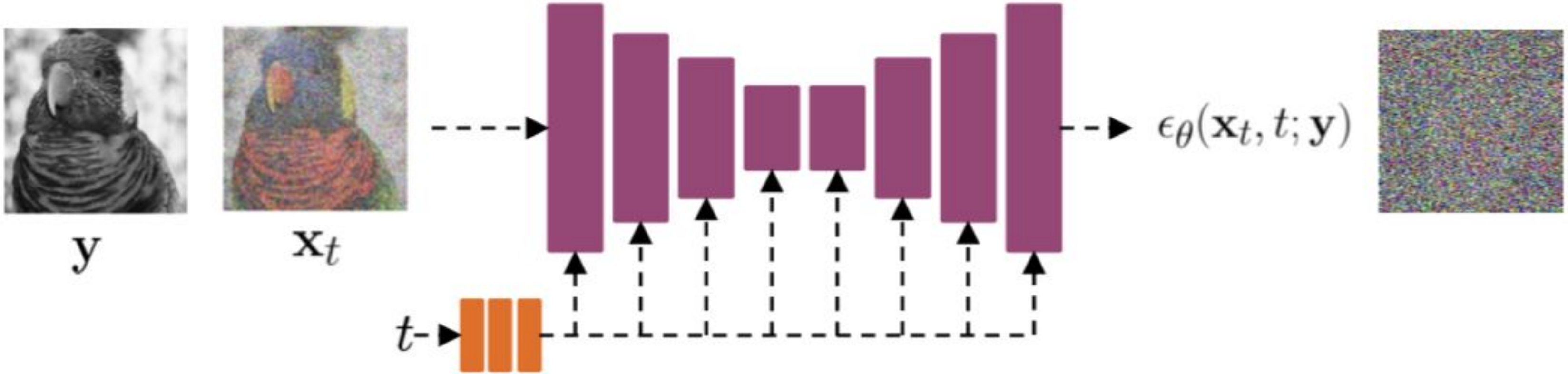
Super-resolution



Saharia et al., Image Super-Resolution via Iterative Refinement, 2021

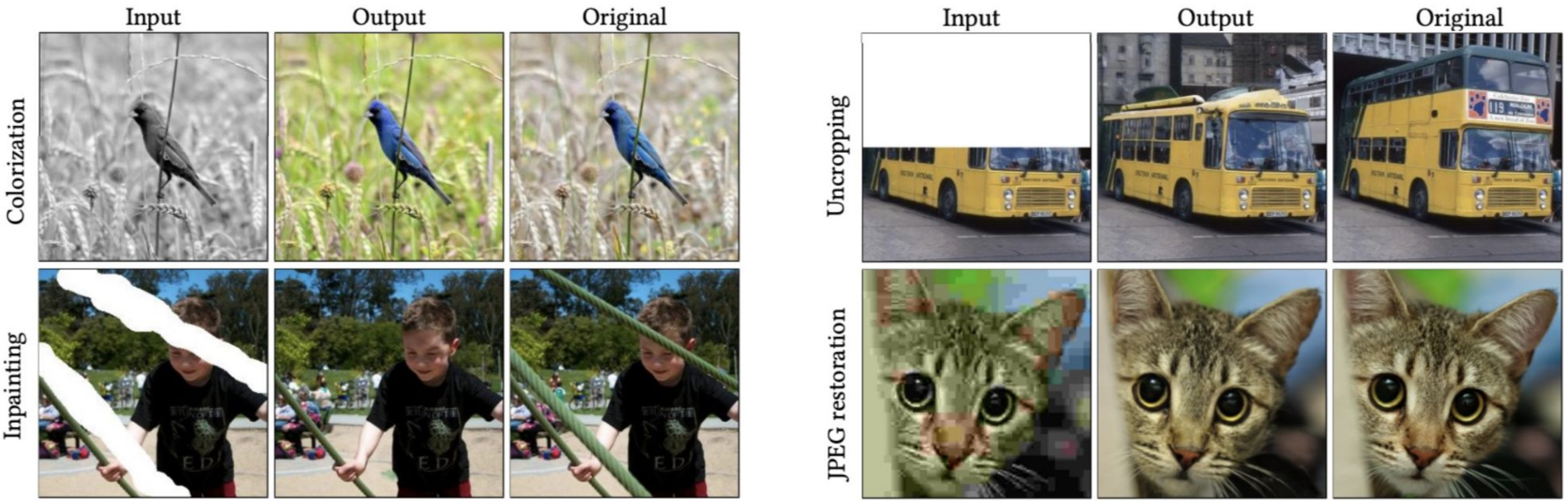
Image-to-image translation

- Image-to-image translation applications can be considered as training p(x | y) where y is the input image
- The conditional score is simply a U-Net with x_t and y (input image) concatenated



Saharia et al., Palette: Image-to-Image Diffusion Models, 2022

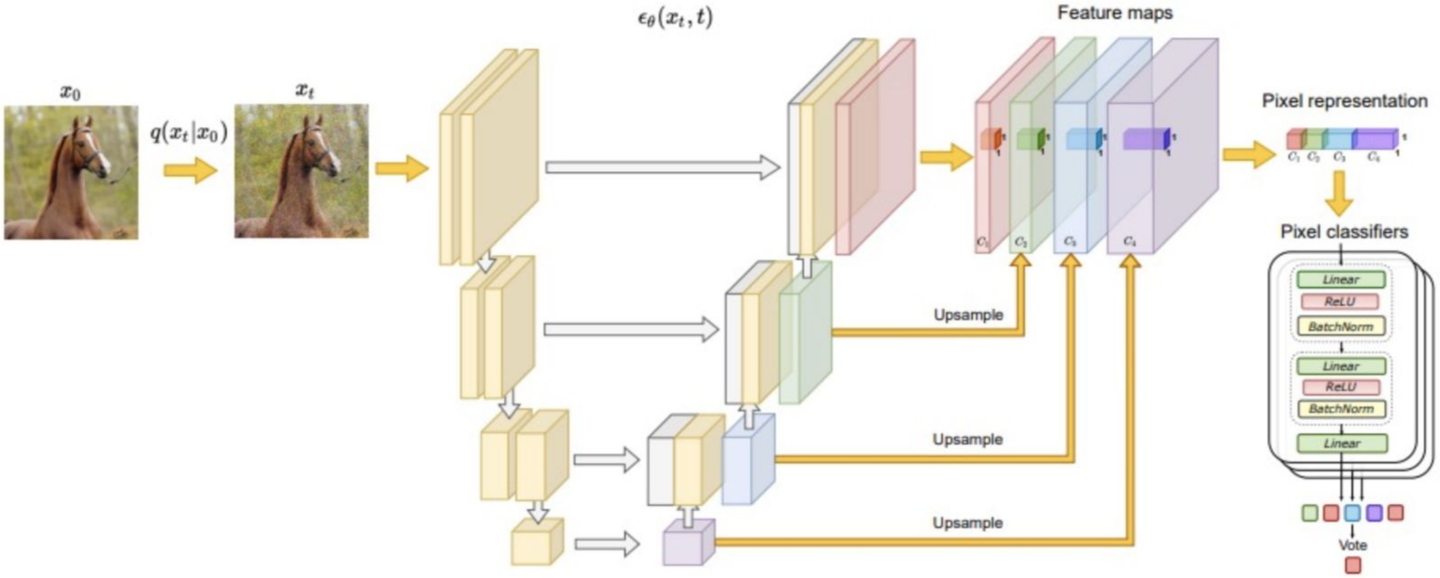
Image-to-image translation



Saharia et al., Palette: Image-to-Image Diffusion Models, 2022

Semantic segmentation

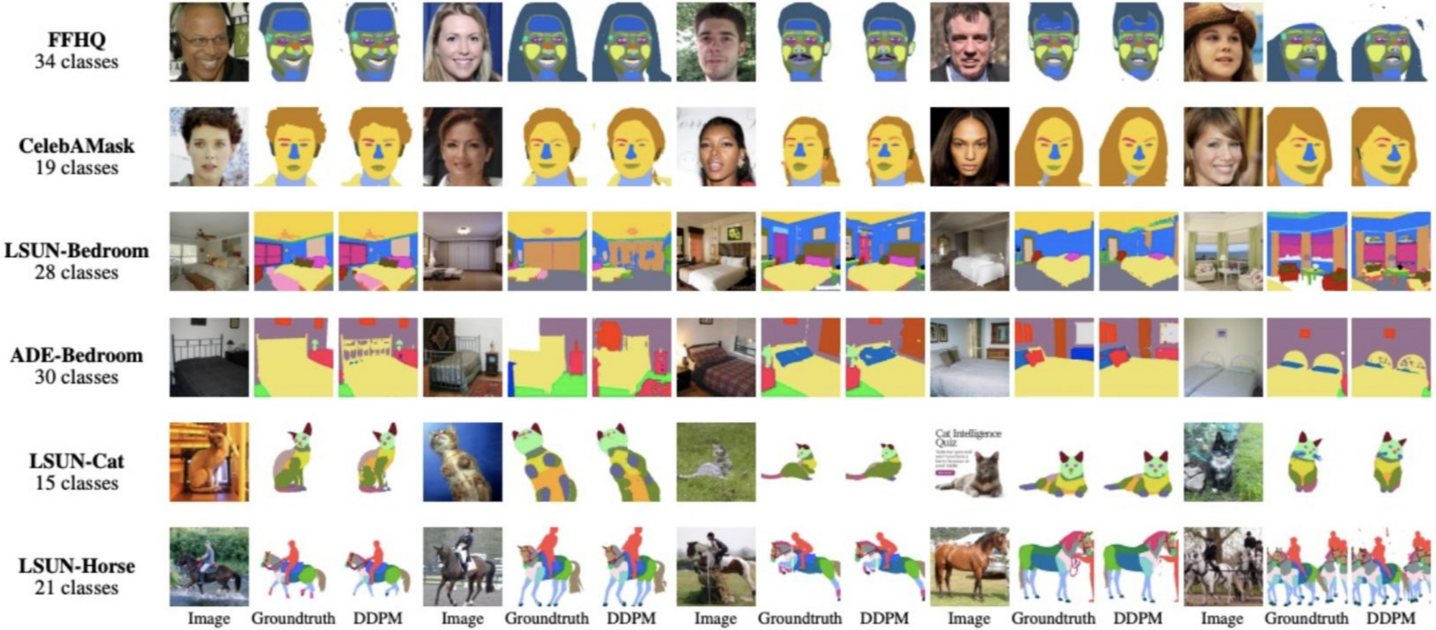
- Use representation learned from diffusion models for downstream applications such as semantic segmentation



Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022

Semantic segmentation

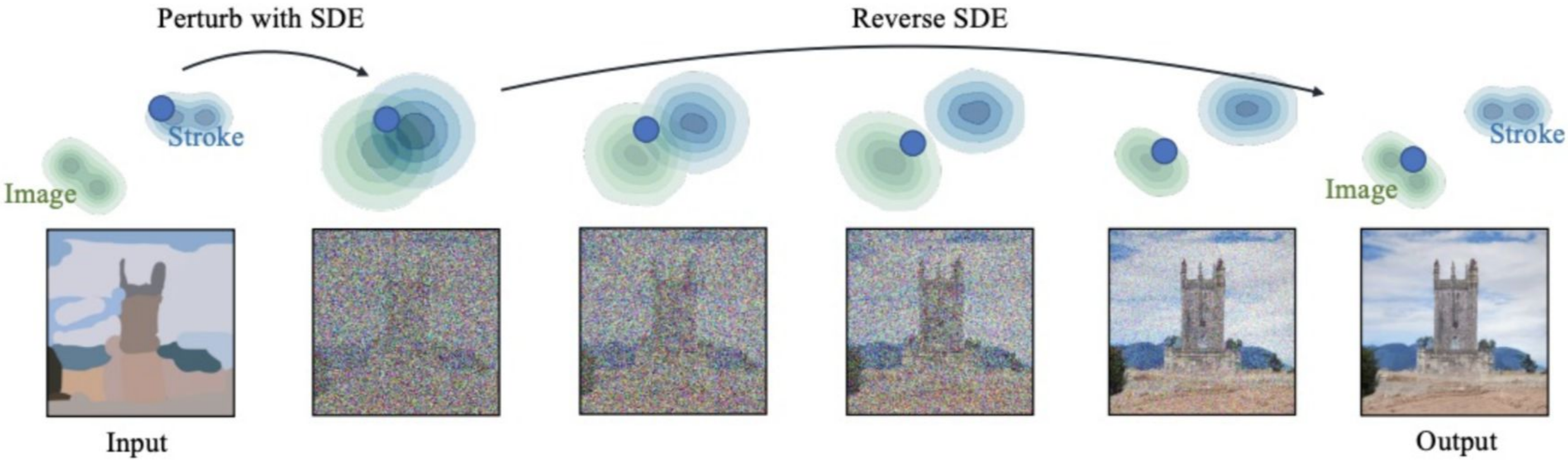
- The experimental results show that the proposed method outperforms Masked Autoencoders, GAN and VAE-based models



Baranchuk et al., Label-Efficient Semantic Segmentation with Diffusion Models, ICLR 2022

Image editing

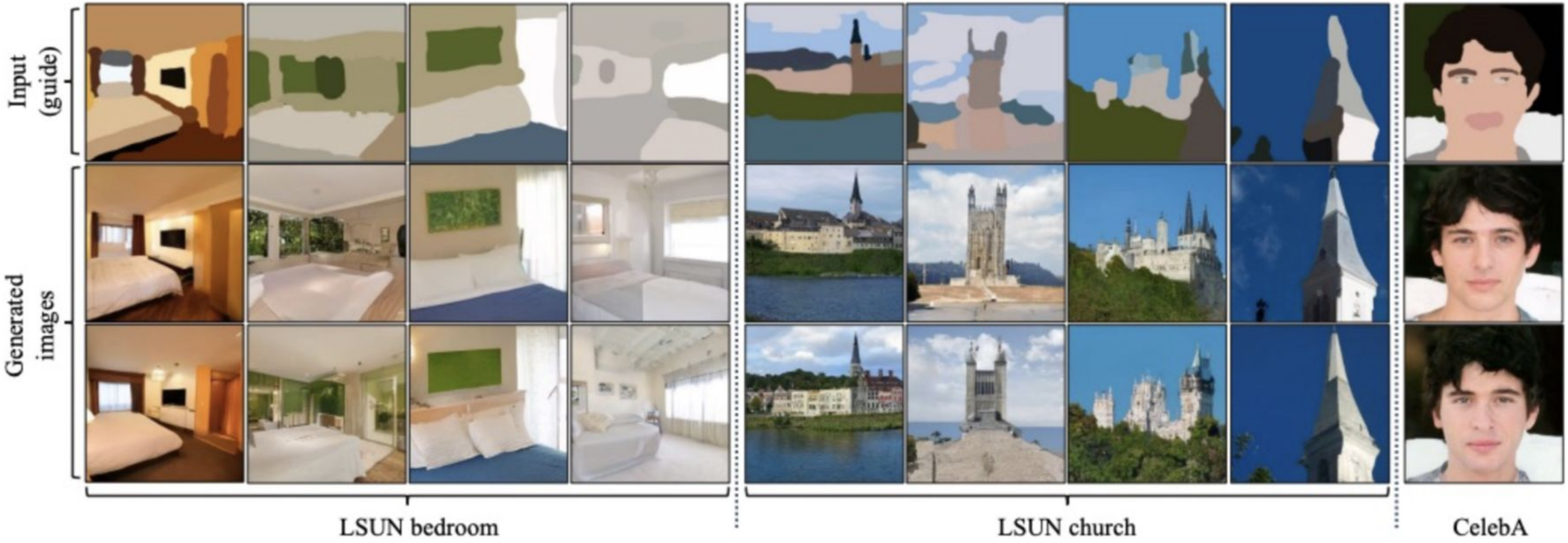
- Forward diffusion brings two distributions close to each other



Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022

Image editing

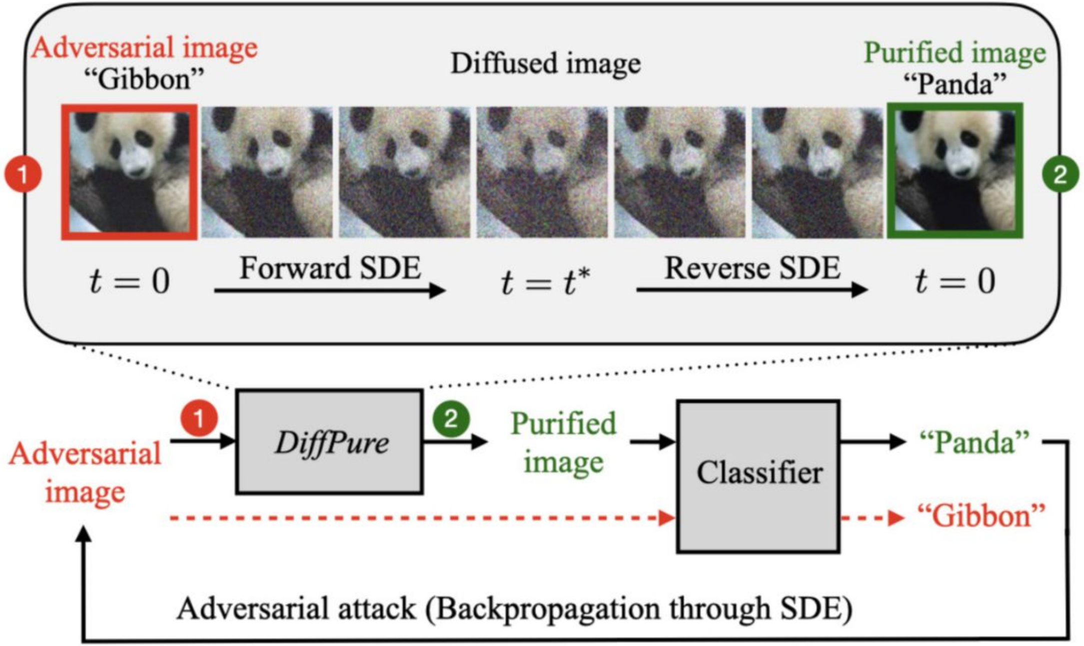
- Generation examples



Meng et al., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, ICLR 2022

Adversarial purification

- Reserve image semantics during diffusion process



Nie et al., Diffusion Models for Adversarial Purification, ICML 2022

Video generation

- Video Generation Tasks:
  - Unconditional generation, future prediction, past prediction, infilling

Samples from a text-conditioned video diffusion model, conditioned on the string *fireworks*.

Ho et al., "Video Diffusion Models", arXiv, 2022
Harvey et al., "Flexible Diffusion Modeling of Long Videos", arXiv, 2022
Yang et al., "Diffusion Probabilistic Modeling for Video Generation", arXiv, 2022
Höppe et al., "Diffusion Models for Video Prediction and Infilling", arXiv, 2022
Voleti et al., "MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation", arXiv, 2022

Lots of literatures …

- Review of latest generative diffusion modeling papers:
  - https://scorebasedgenerativemodeling.github.io/

Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

Challenges

- Time efficiency:
  - Sampling process for diffusion model is still slow
  - Improve training and sampling time efficiency
  - New diffusion process? Consistency model?
- Memory efficiency:
  - How to handle high-dimensional high-resolution images in real word?
  - Pixel or latent diffusion model?
  - 3D convolution or 2D convolution + attention layer?
- Data efficiency:
  - How many data is sufficient for modeling data distribution?
  - Out-of-distribution problems

Challenges

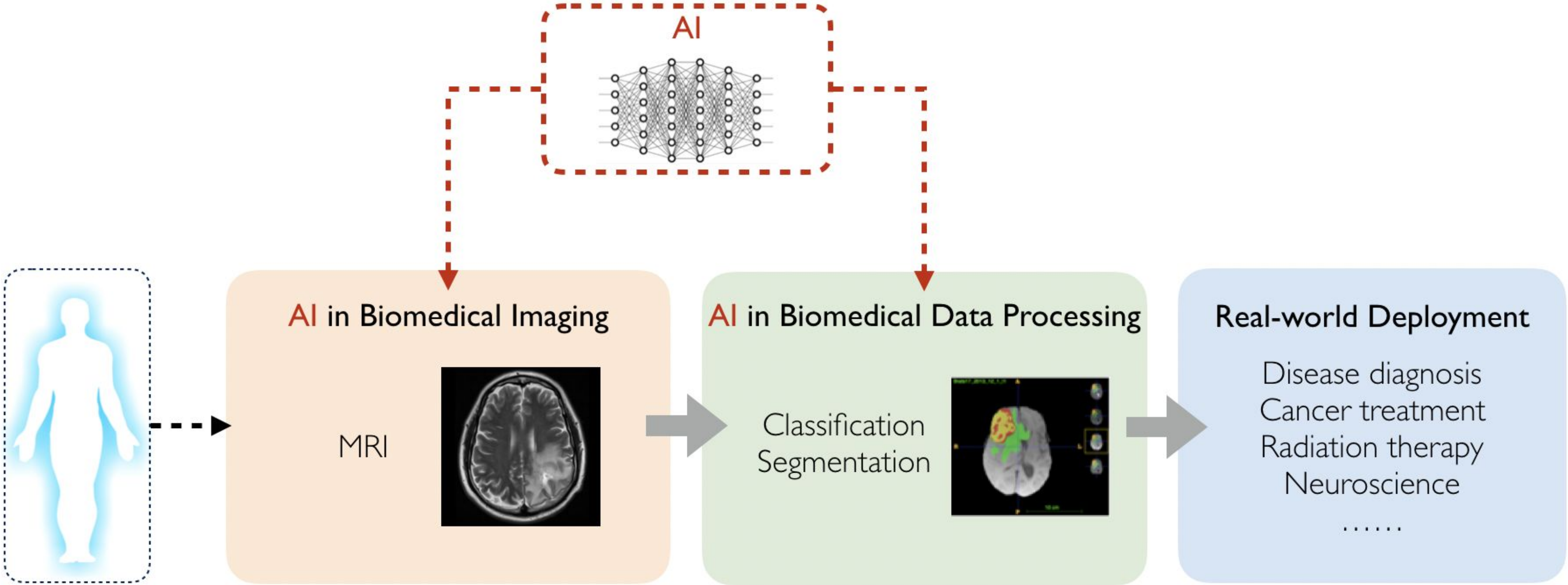
- How can diffusion models help with discriminative applications?
- How to do latent-space semantic manipulations in diffusion models?
- What are the best network architectures for diffusion models?
- How to apply diffusion models to other data types?
- Can we better solve applications that were previously addressed by GANs and other generative models?

Today's agenda

- Generative diffusion models
  - Score-based generative models
  - Denoising diffusion probabilistic models
- Image generation
- Inverse problem solving
- Other applications
- Challenges

Next lecture

- Part I: AI in biomedical imaging
- Part II: AI in biomedical data processing



Next lecture

- Medical image analysis

Date	Lecture #	Topic	Papers	Instructor / Presenter
Tue 8/29	1	Introduction and course overview		Liyue Shen
Thu 8/31	2	Biomedical imaging with deep learning **[Fundamental]**		Liyue Shen
Tue 9/5	3	Implicit neural representation learning **[Advanced]**		Liyue Shen
Thu 9/7	4	Generative diffusion models **[Advanced]**		Liyue Shen
Tue 9/12	5	Medical image analysis **[Fundamental]**		Liyue Shen
Thu 9/14	6	Multimodal foundation models **[Advanced]**		Liyue Shen
Mon 9/18		Drop/add deadline for full term classes		
Tue 9/19	7	Implicit neural representation learning		
Thu 9/21	8	Implicit neural representation learning		
Tue 9/26	9	Implicit neural representation learning		
Thu 9/28	10	Implicit neural representation learning		
Tue 10/3	11	Generative diffusion models		
Thu 10/5	12	Generative diffusion models		
Tue 10/10	13	Generative diffusion models		
Thu 10/12	14	Generative diffusion models		
Tue 10/17		No class (fall study break)		
Thu 10/19	15	Self-supervised learning		
Tue 10/24	16	Self-supervised learning		
Thu 10/26	17	Multimodal learning		
Tue 10/31	18	Multimodal learning		
Thu 11/2	19	Transformer and LLM		
Tue 11/7	20	Transformer and LLM		