Diffusion Model Review

CAO Hanqun

Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Outlines

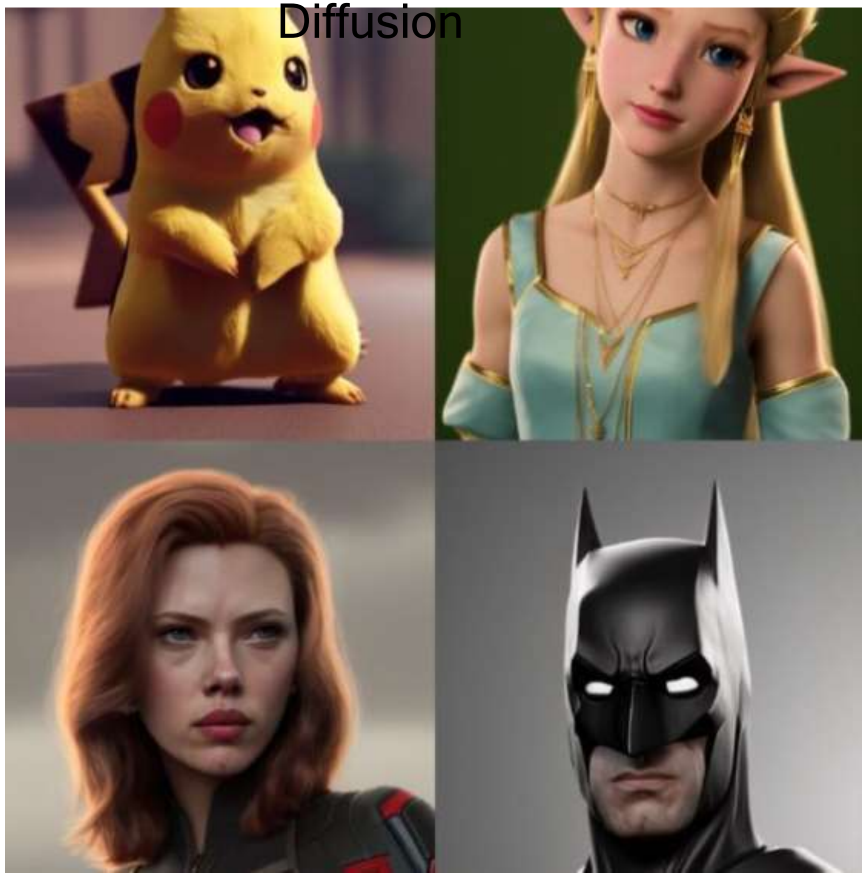
- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Diffusion Model is Striking Right Now!

Stable Diffusion



Midjourney

Explosive Growth of Diffusion Model

Denoising diffusion probabilistic models

J Ho, A Jain, P Abbeel - Advances in neural information …, 2020 - proceedings.neurips.cc

… This paper presents progress in diffusion probabilistic models [53]. A diffusion probabilistic model (which we will call a "diffusion model" for brevity) is a parameterized Markov chain …

☆ 保存 99 引用 被引用次数：3640 相关文章 所有 6 个版本 导入BibTeX »

Score-based generative modeling through stochastic differential equations

Y Song, J Sohl-Dickstein, DP Kingma, A Kumar… - arXiv preprint arXiv …, 2020 - arxiv.org

… Figure 1: Solving a reversetime SDE yields a score-based generative model. Transforming … a continuous-time SDE. This SDE can be reversed if we know the score of the distribution at …

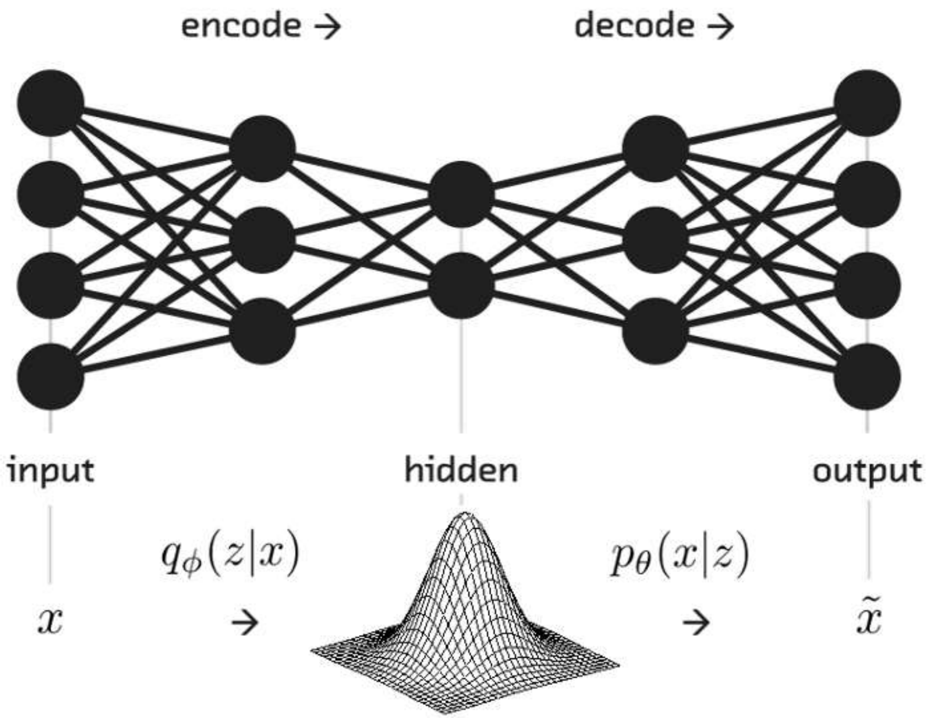
☆ 保存 99 引用 被引用次数：1580 相关文章 所有 5 个版本 导入BibTeX »

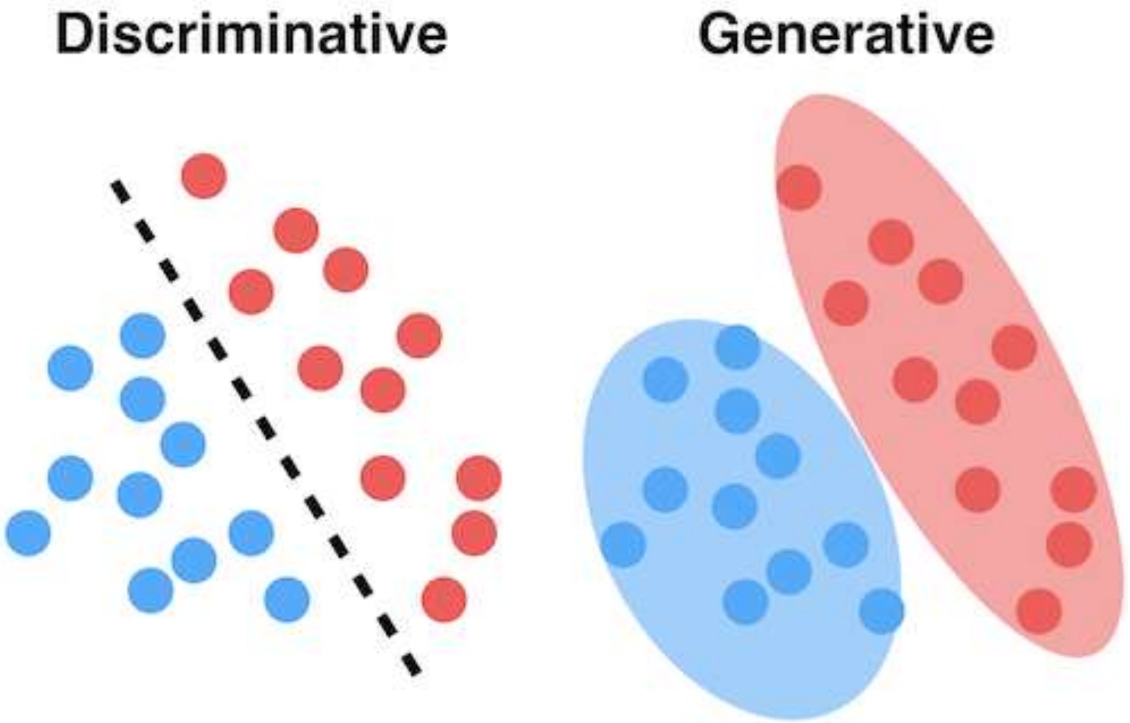
Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithm
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

From Generative Model to Diffusion Model

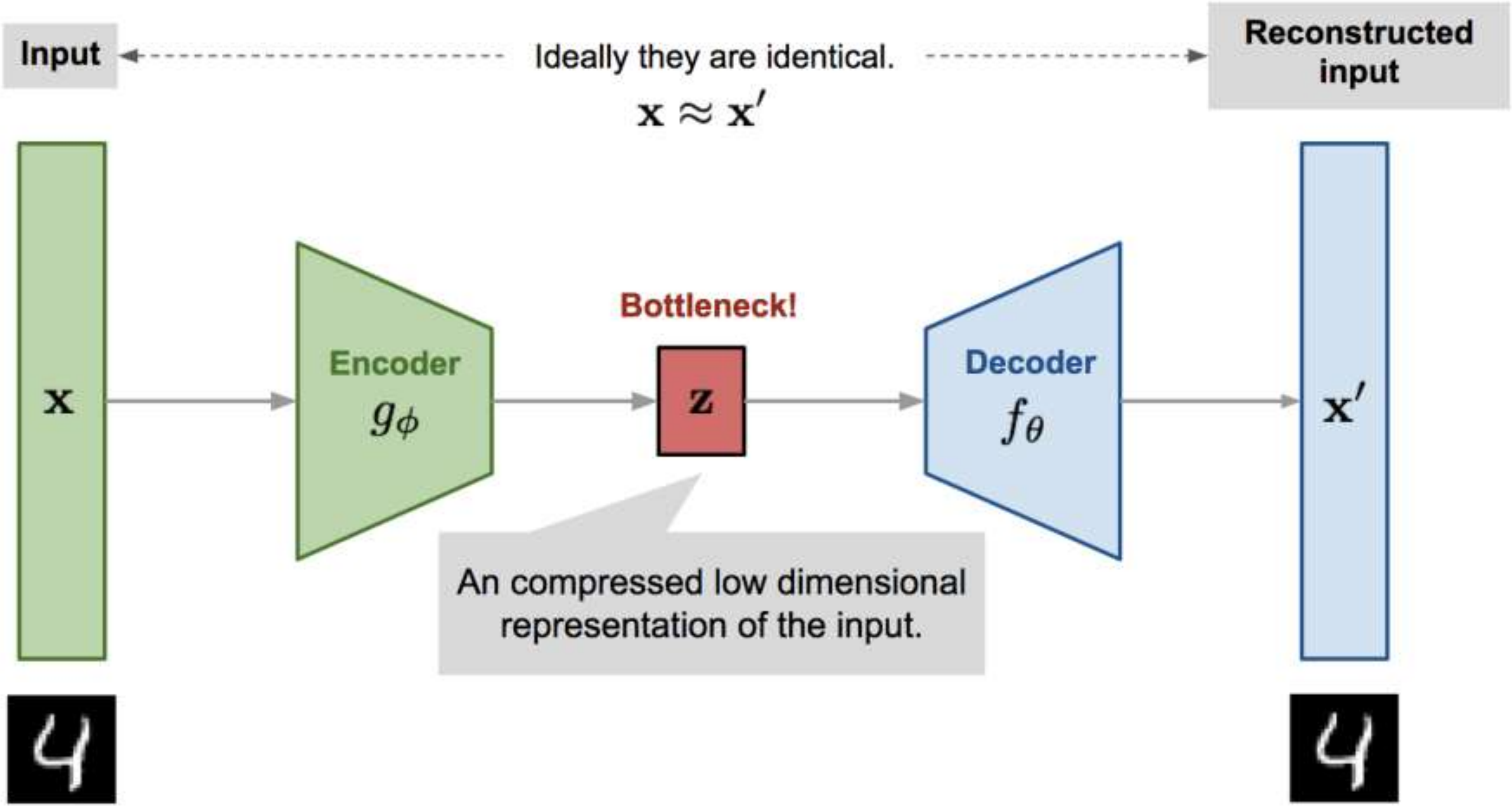
- Given data x, generating samples following distribution p(x)





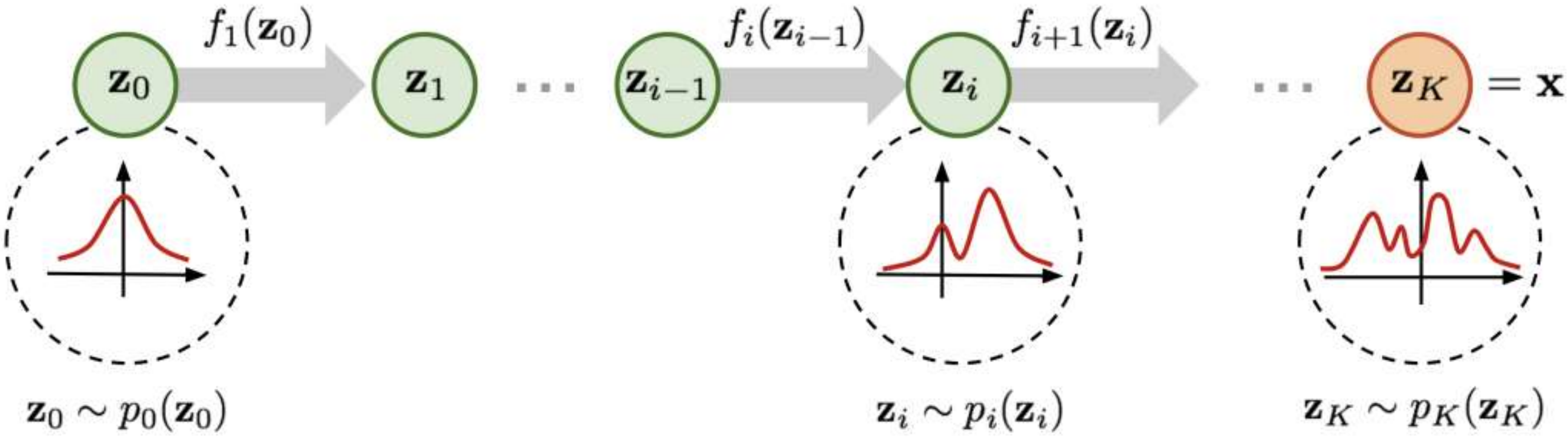
From VAE to NF, to Diffusion Model

How to accurate and efficiently express the distribution remains to be a challenge



VAE suffers from an information loss when conducting encoding

From VAE to NF, to Diffusion Model

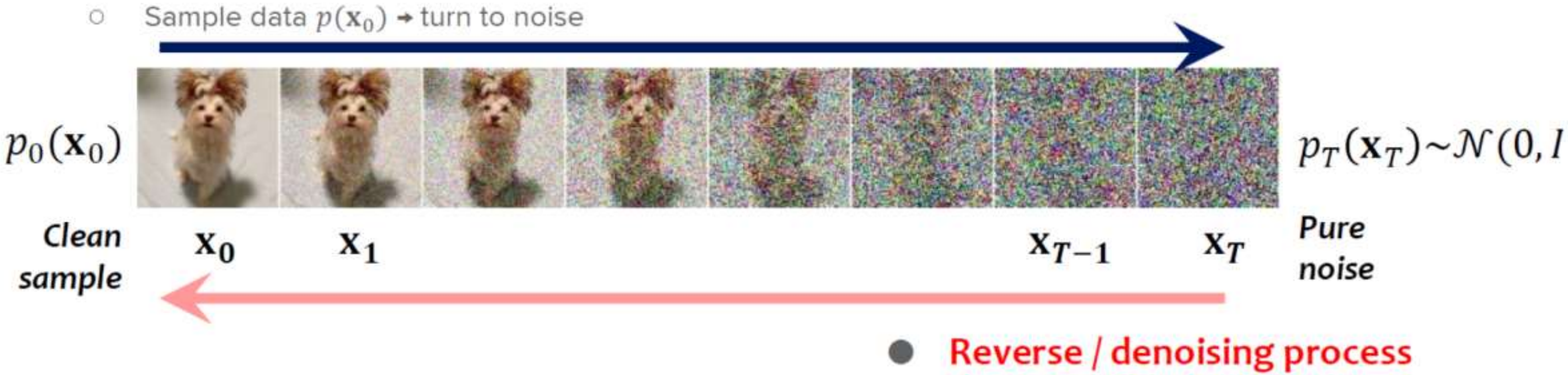


Normalizing Flow (NF) suffers from complicated modeling and training

From VAE to NF, to Diffusion Model

No information loss -> equal-dimension transformation

Efficient Modeling -> Markovian process



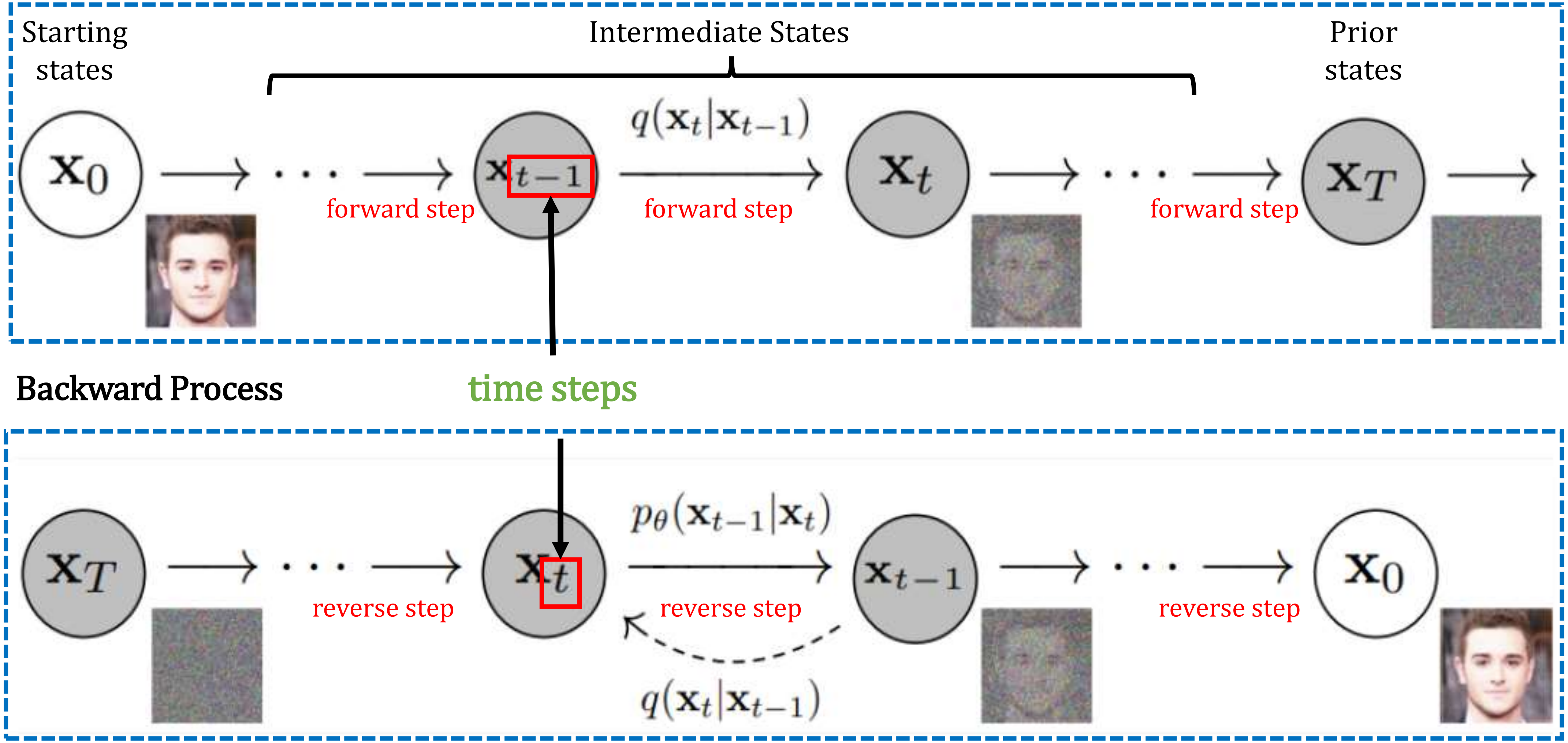
Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithms
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Basic Definitions

Forward Process

Backward Process



Landmark Works: DDPM

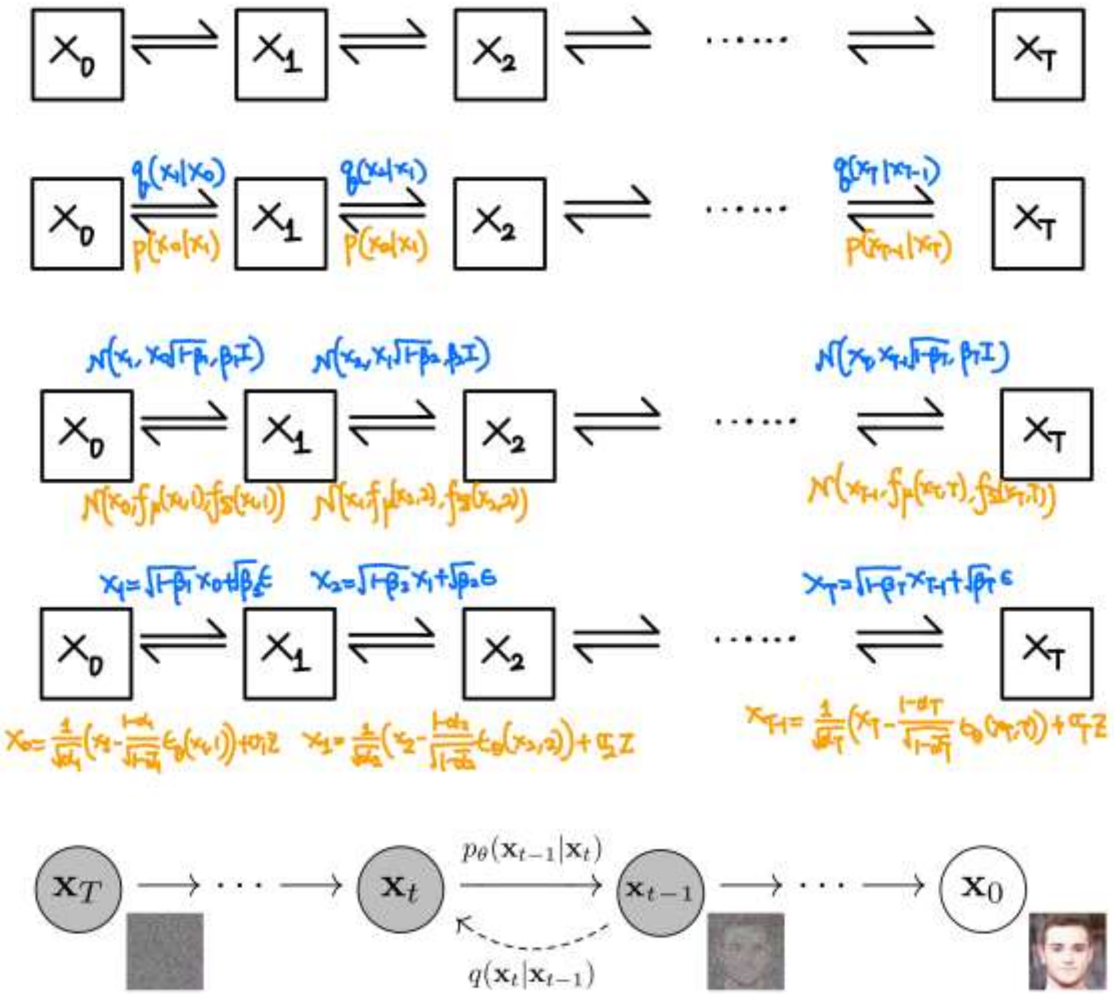


Figure 2: The directed graphical model considered in this work.

Algorithm 1 Training

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

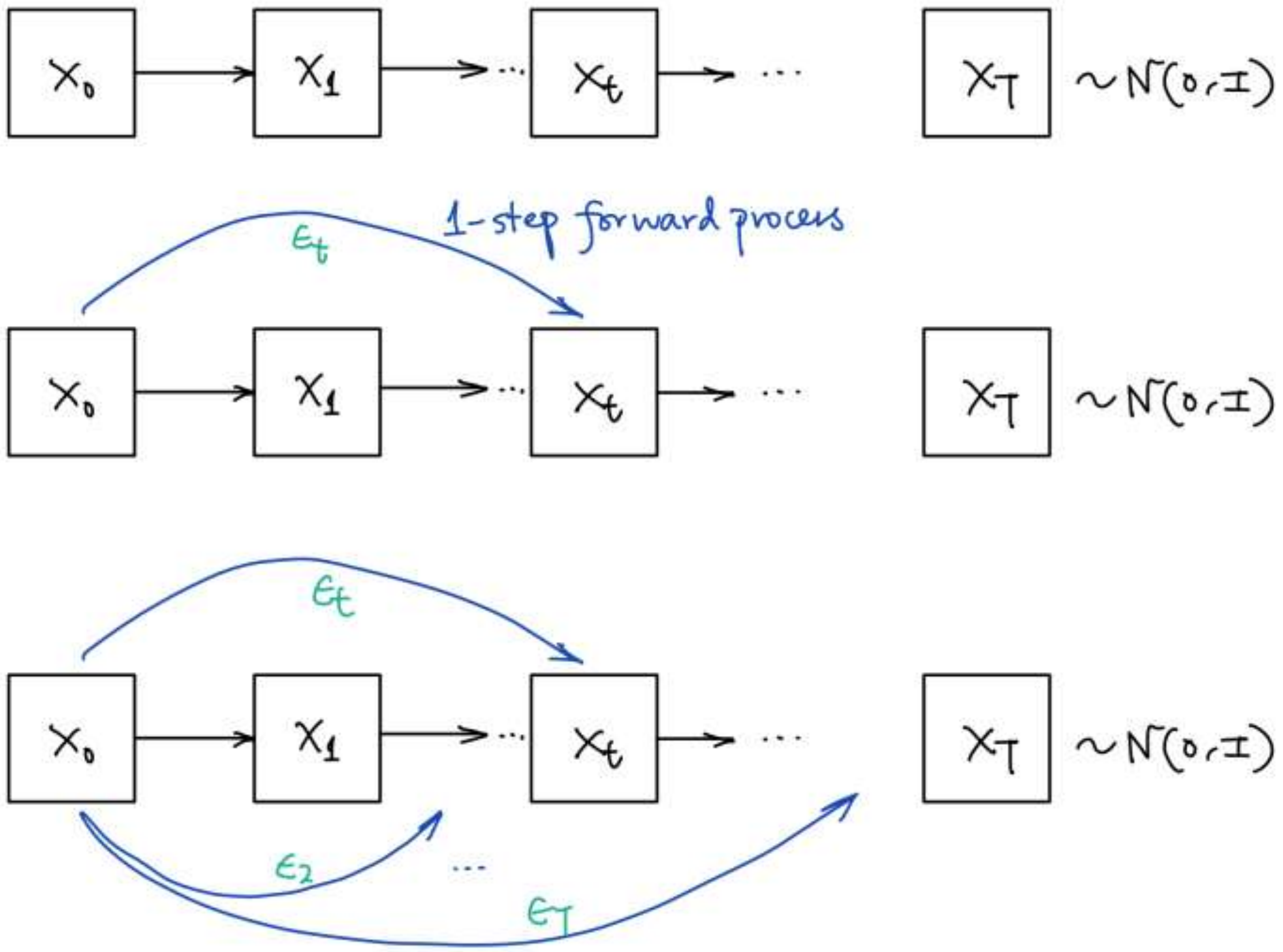
$$\boxed{\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2}$$

6: **until** converged

Algorithm 2 Sampling

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

Landmark Works: DDPM

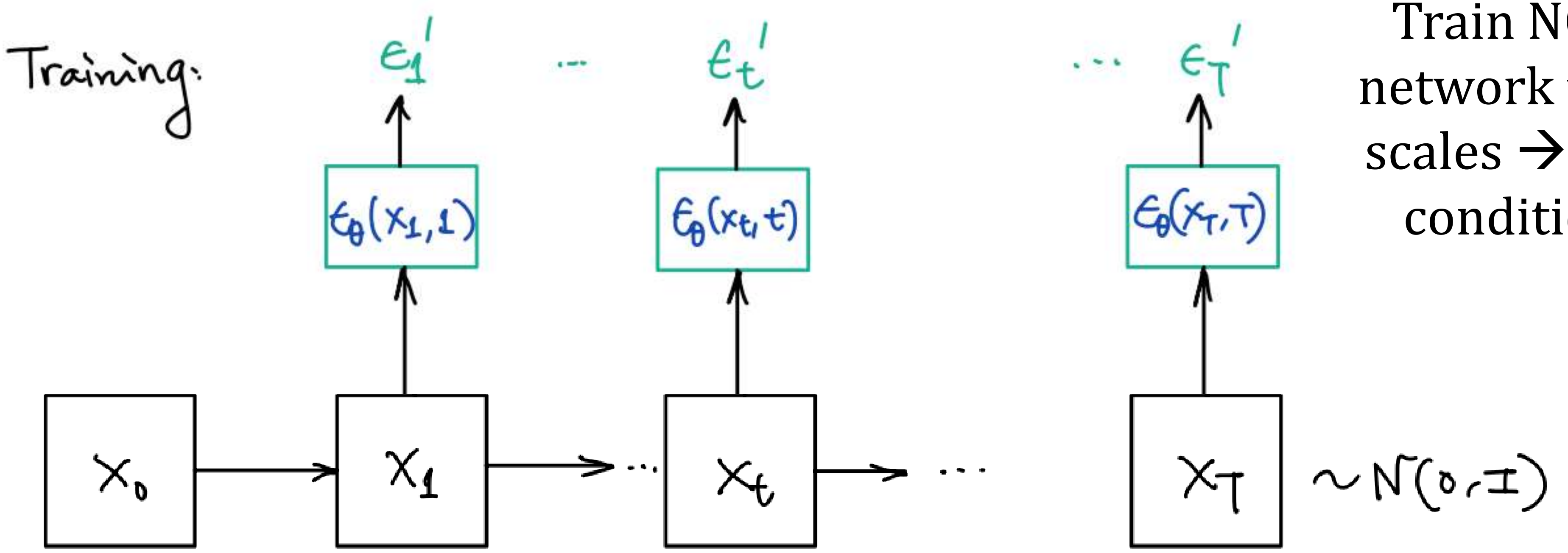


Initialization

Forward step with random noise

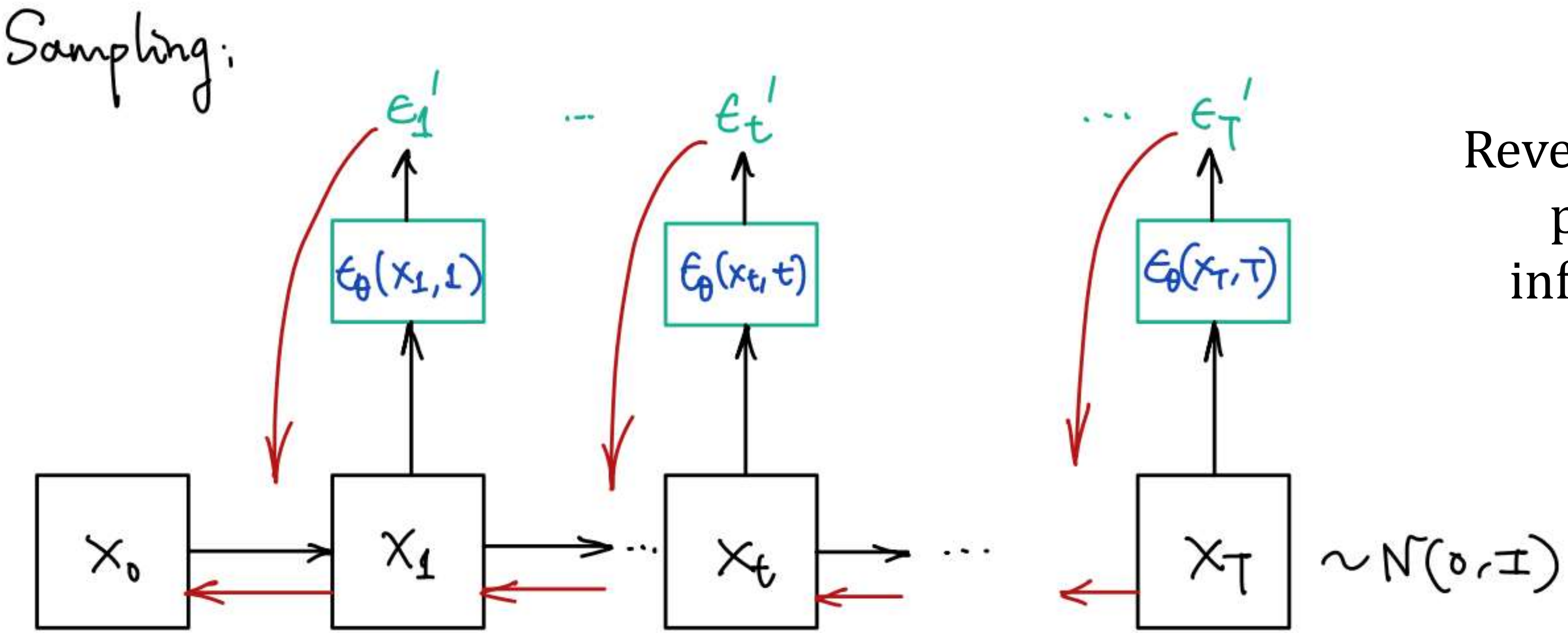
Forward steps with diverse random noise

Landmark Works: DDPM



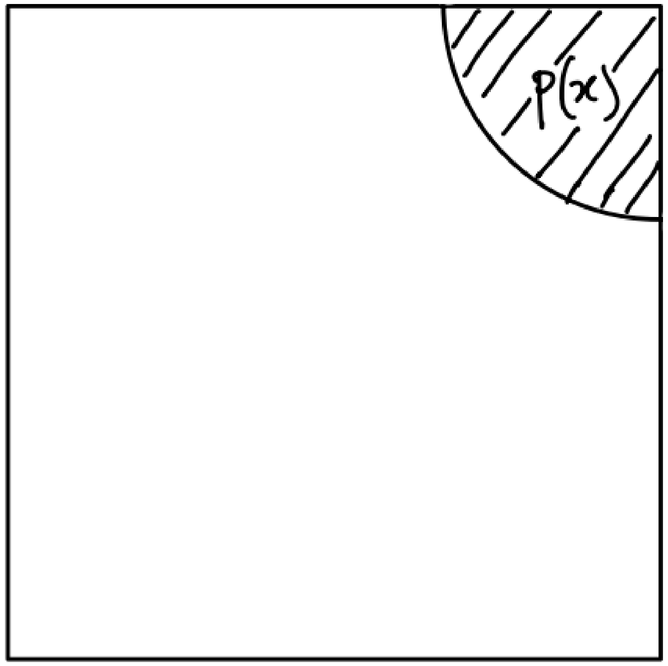
Train NOISE PREDICTION network to fit different noise scales → time is treated as a condition to the network

Landmark Works: DDPM

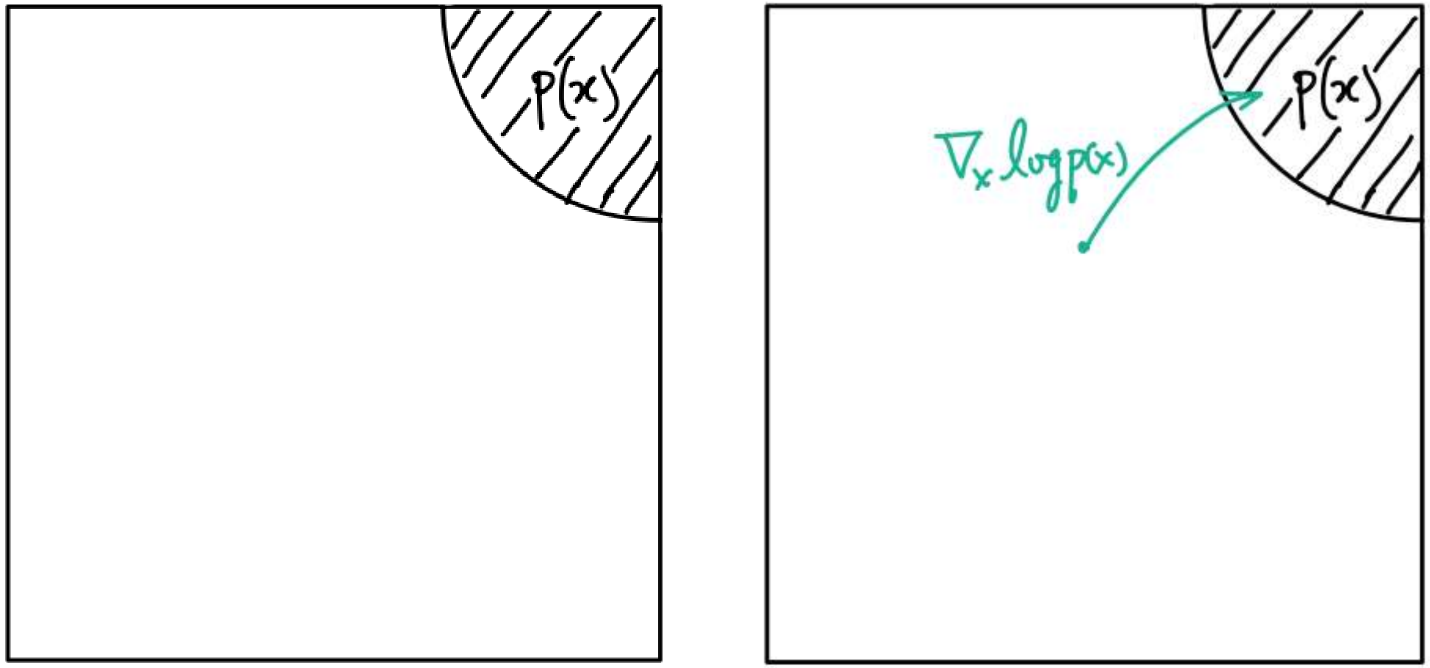


Reverse the forward process with inference noises

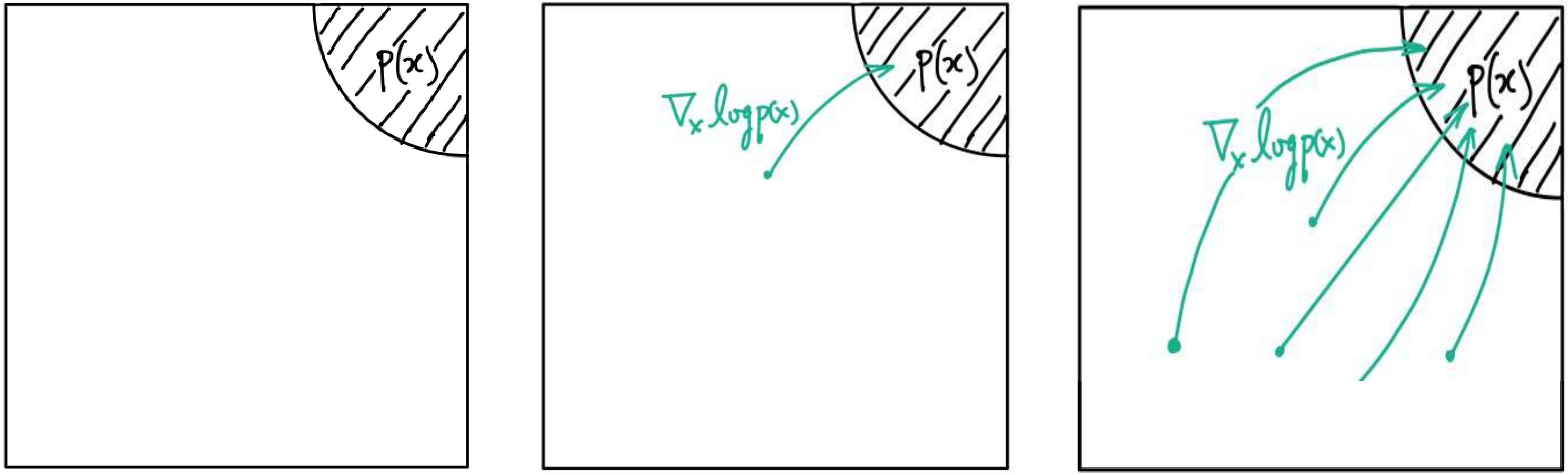
Landmark Works: Score Network



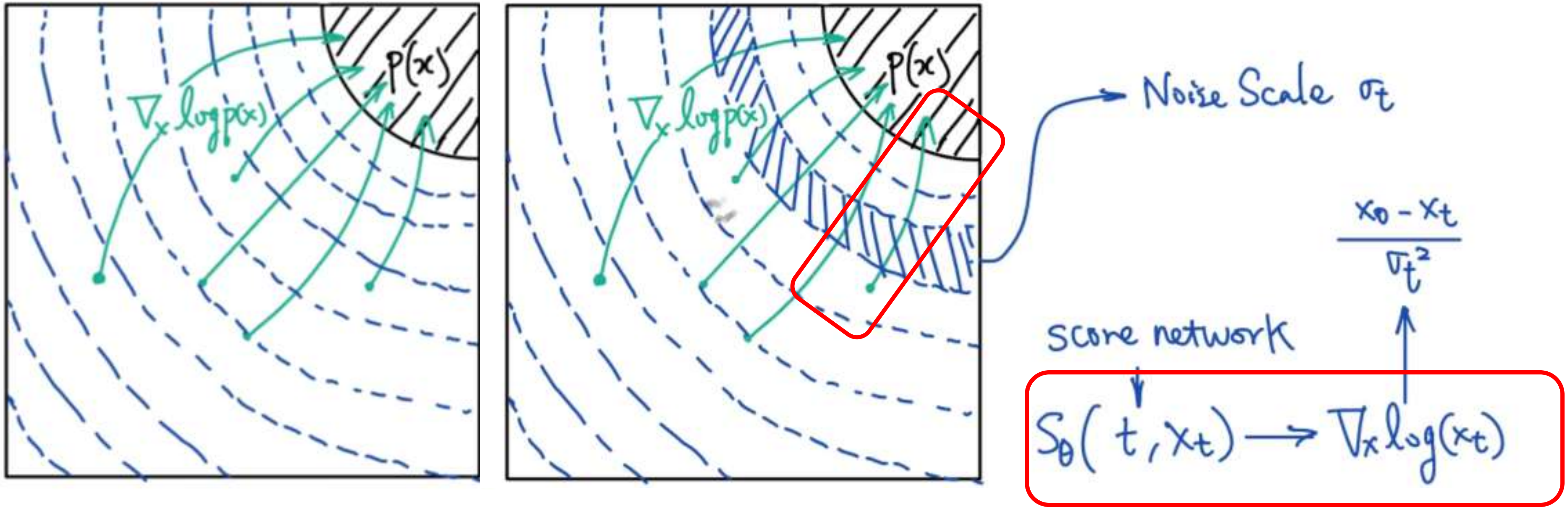
Landmark Works: Score Network



Landmark Works: Score Network



Landmark Works: Score Network



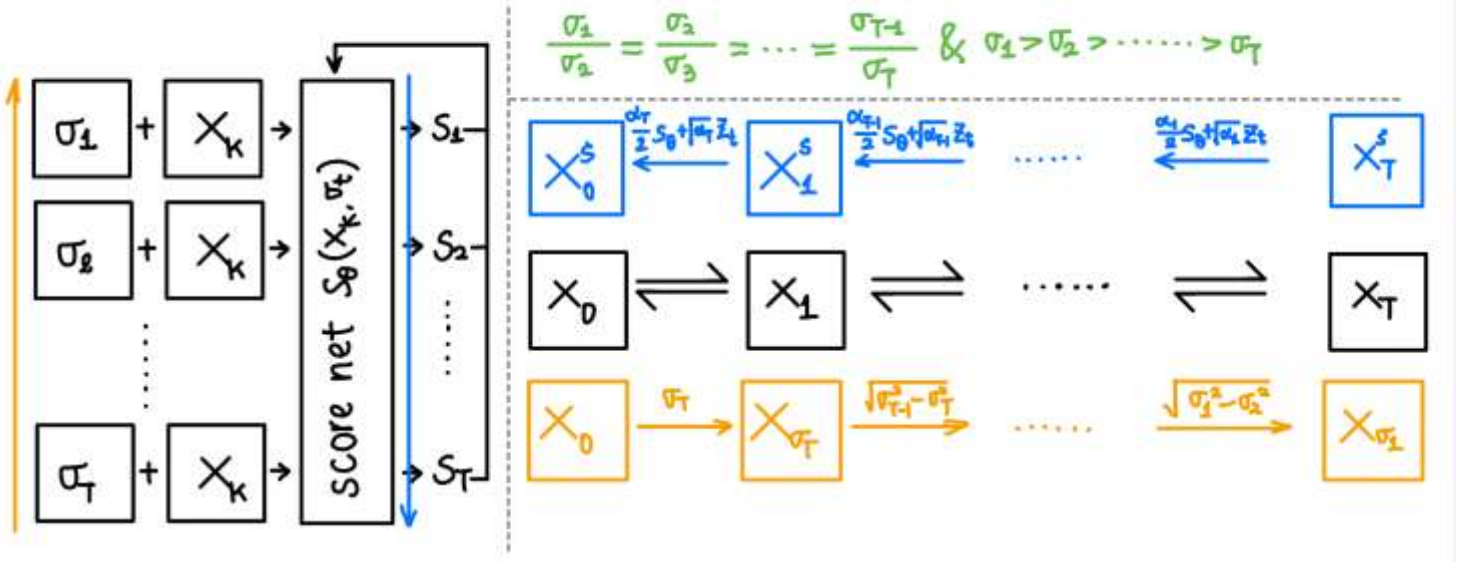
Landmark Works: Score Network

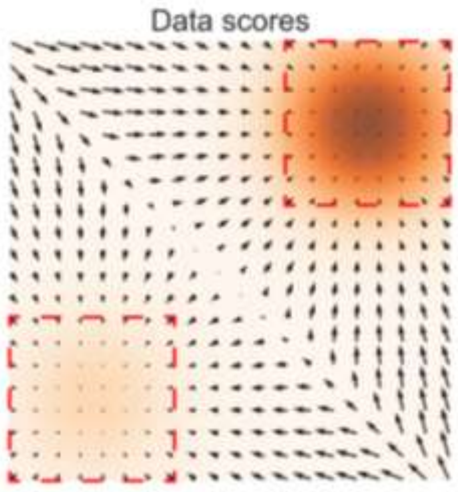
Obtaining distributions by estimating its gradients

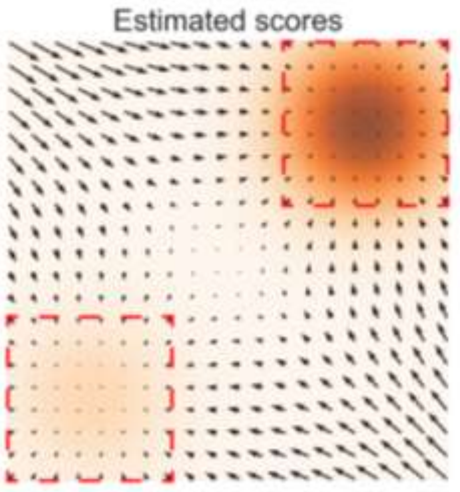
Score

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

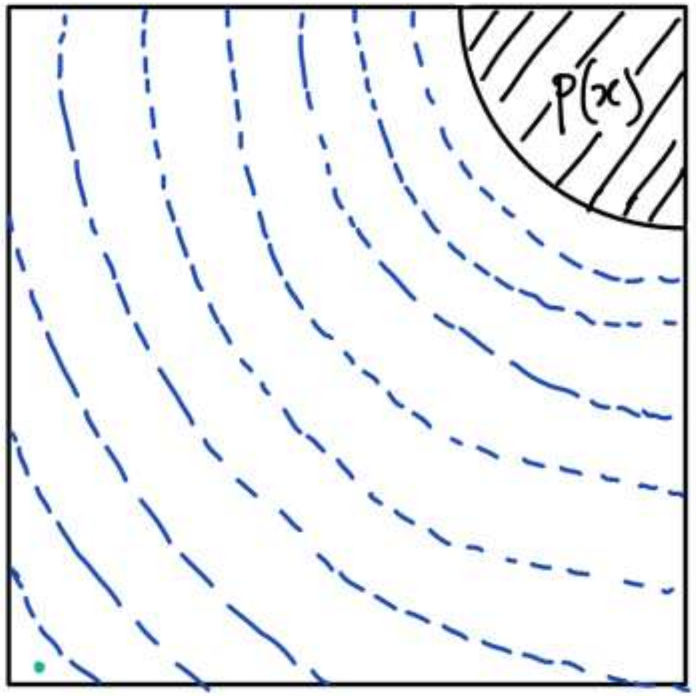
$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \boxed{\epsilon \nabla_\mathbf{x} \log p(\mathbf{x})} + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K,$$

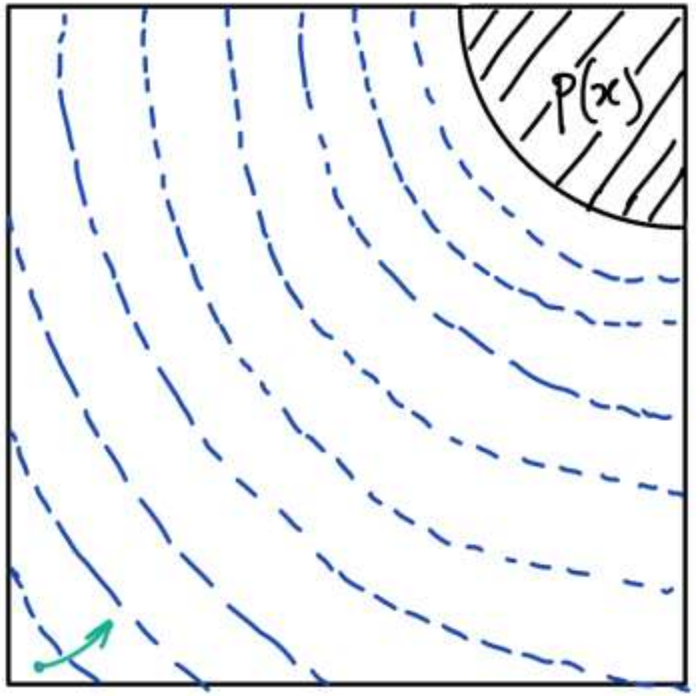




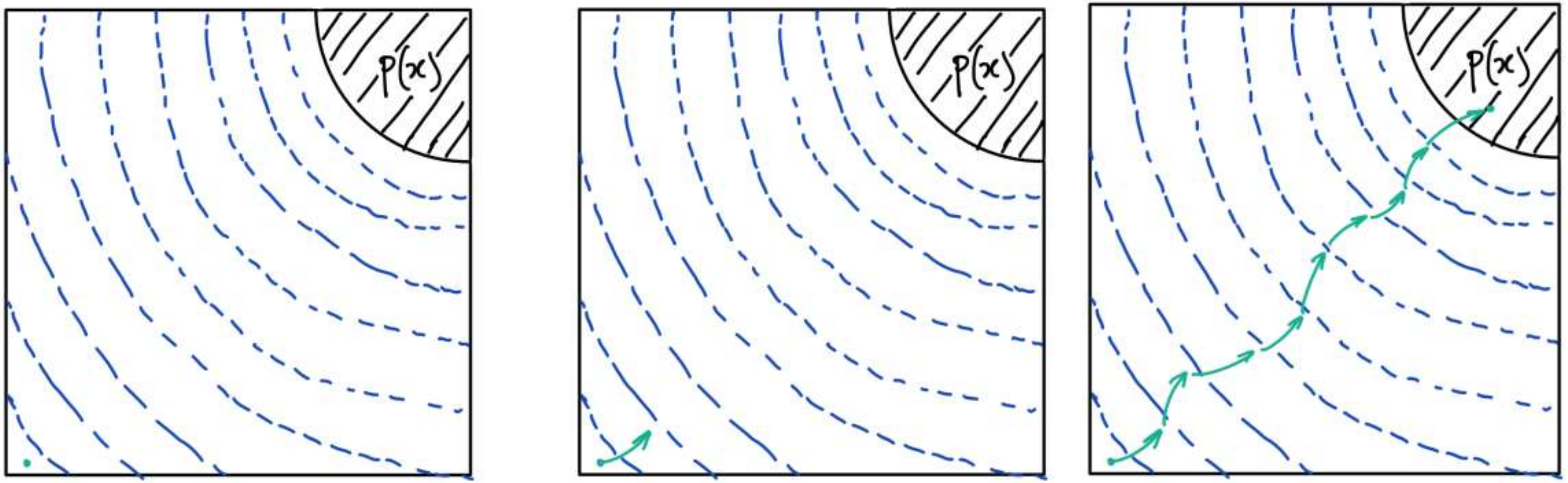


Landmark Works: Score Network



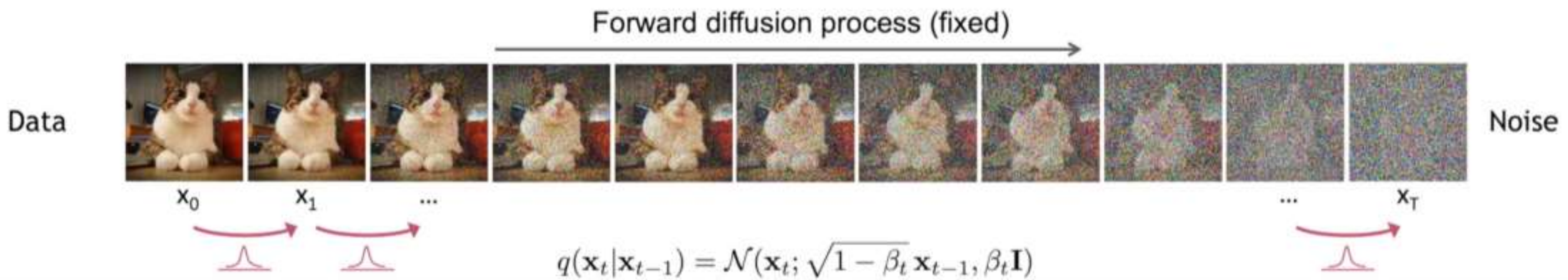


Landmark Works: Score Network



Landmark Works: From discrete to Continuous

Consider the limit of **many small steps**:

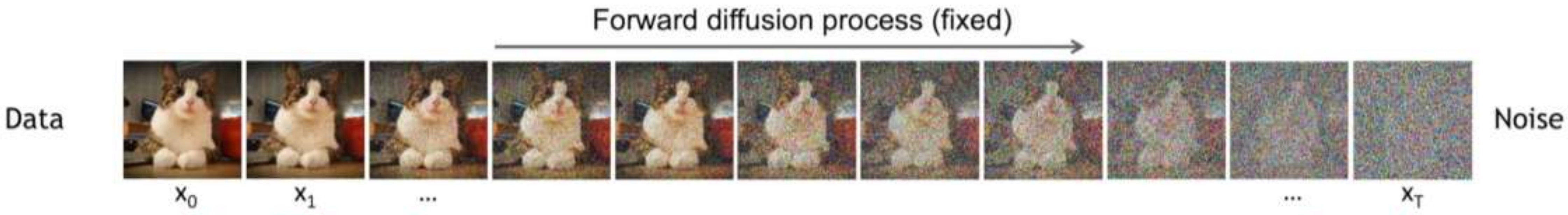


$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

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

Landmark Works: ScoreSDE

Consider the limit of **many small steps:**



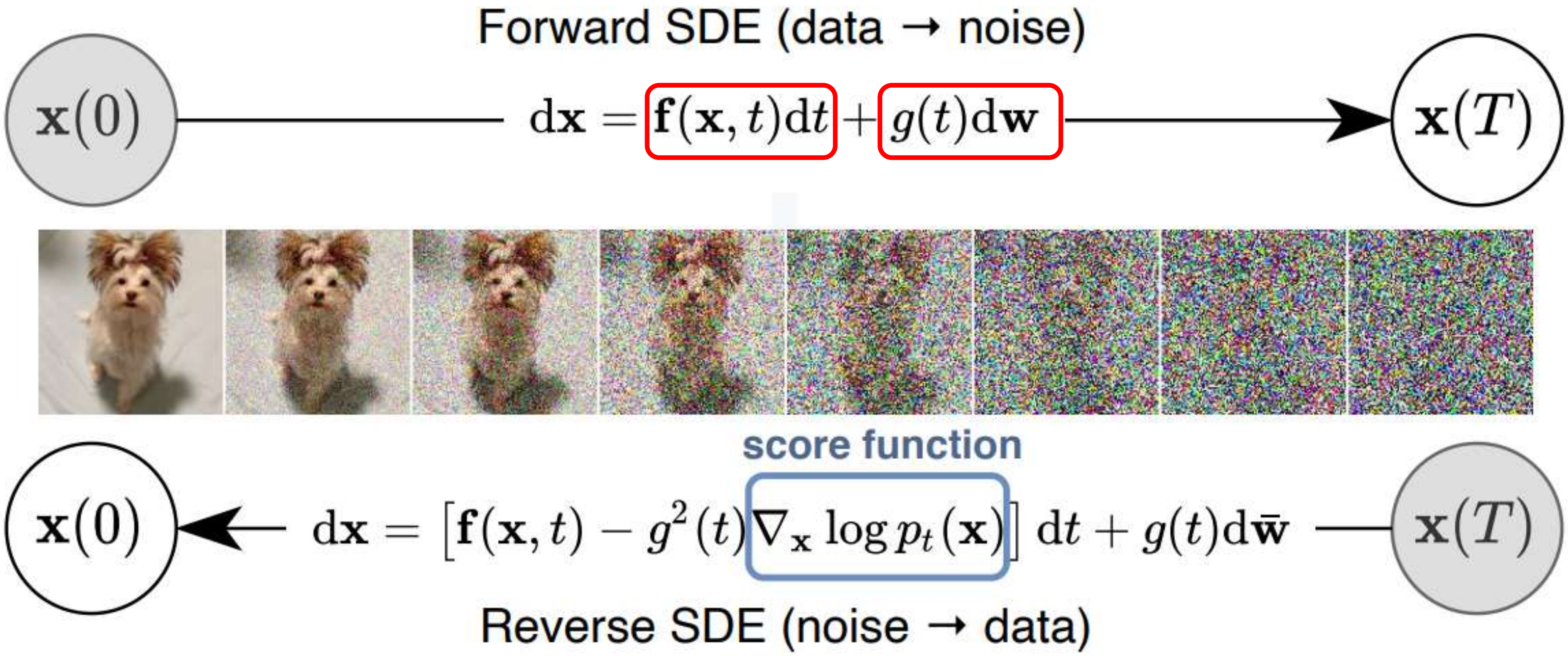
$$\mathbf{x}_t \approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\Downarrow$$

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

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

Landmark Works: ScoreSDE



Landmark Works: Differential Equation Views

$$p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0)) = \begin{cases} \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0), [\sigma^2(t) - \sigma^2(0)]\mathbf{I}\right), & \text{(VE SDE)} \\ \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(s)\mathrm{d}s}\right) & \text{(VP SDE)} \\ \mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}, [1 - e^{-\int_0^t \beta(s)\mathrm{d}s}]^2\mathbf{I}\right) & \text{(sub-VP SDE)} \end{cases}$$

Algorithm 2 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 3 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow (2 - \sqrt{1 - \beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\beta_{i+1}}\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Predictor

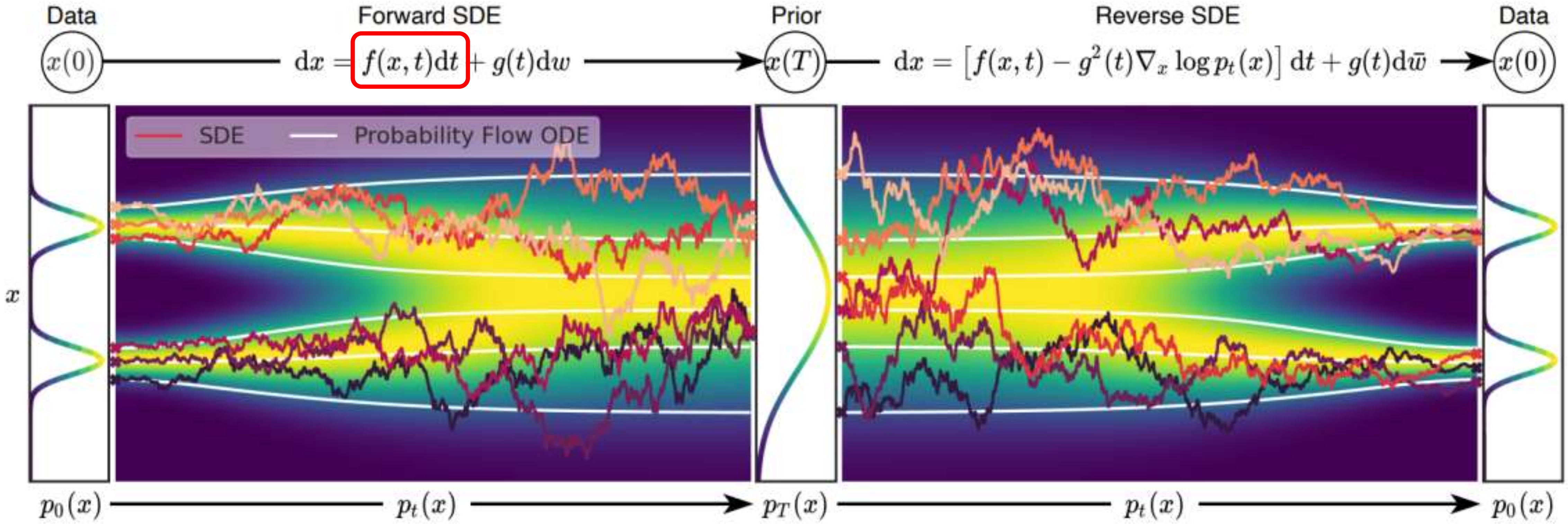
Corrector

-- Sampling along the reverse trajectory is actually finding the numerical solution of differential equations.

Outlines

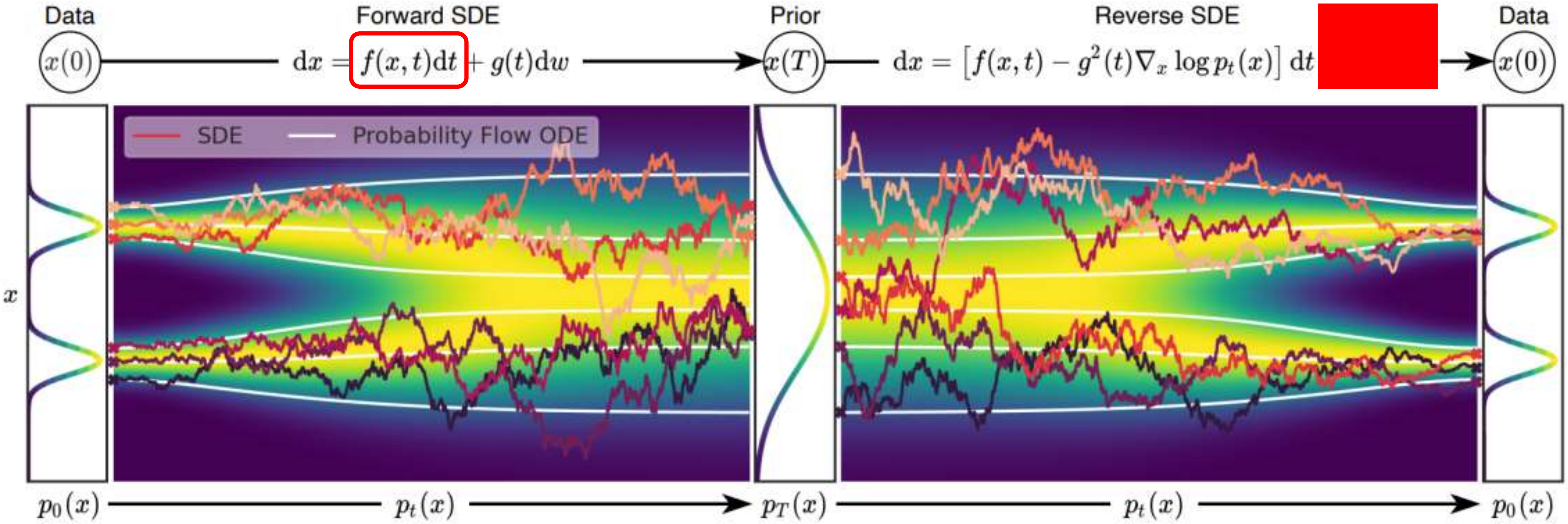
- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithms
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Stochastic or Deterministic?



- SDE: Higher Performance
- ODE: Higher Speed

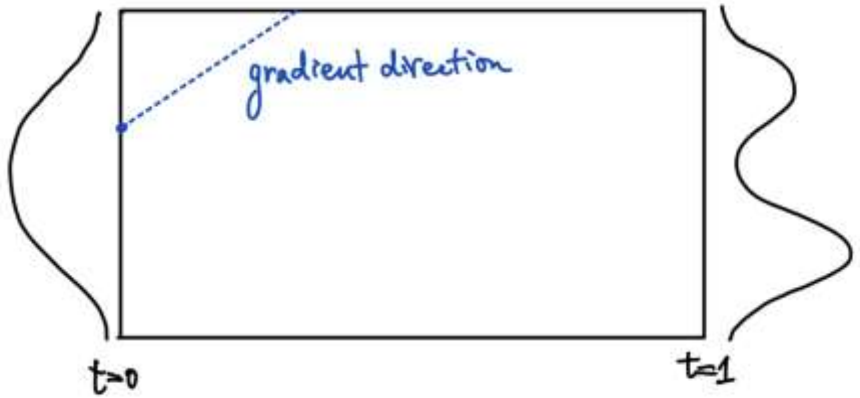
Stochastic or Deterministic?



- SDE: Higher Performance
- ODE: Higher Speed

Stochastic or Deterministic?

- Deterministic sampling is not equivalent to no diversity
  -> there are infinite random points which can be sampled from prior distributions
- Why ODEs are faster?
  -> no randomness leads to larger steps
  -> but the error would accumulate
- Regarding SDE
  -> small steps leads to more steps
  -> random noise for each steps brings less error



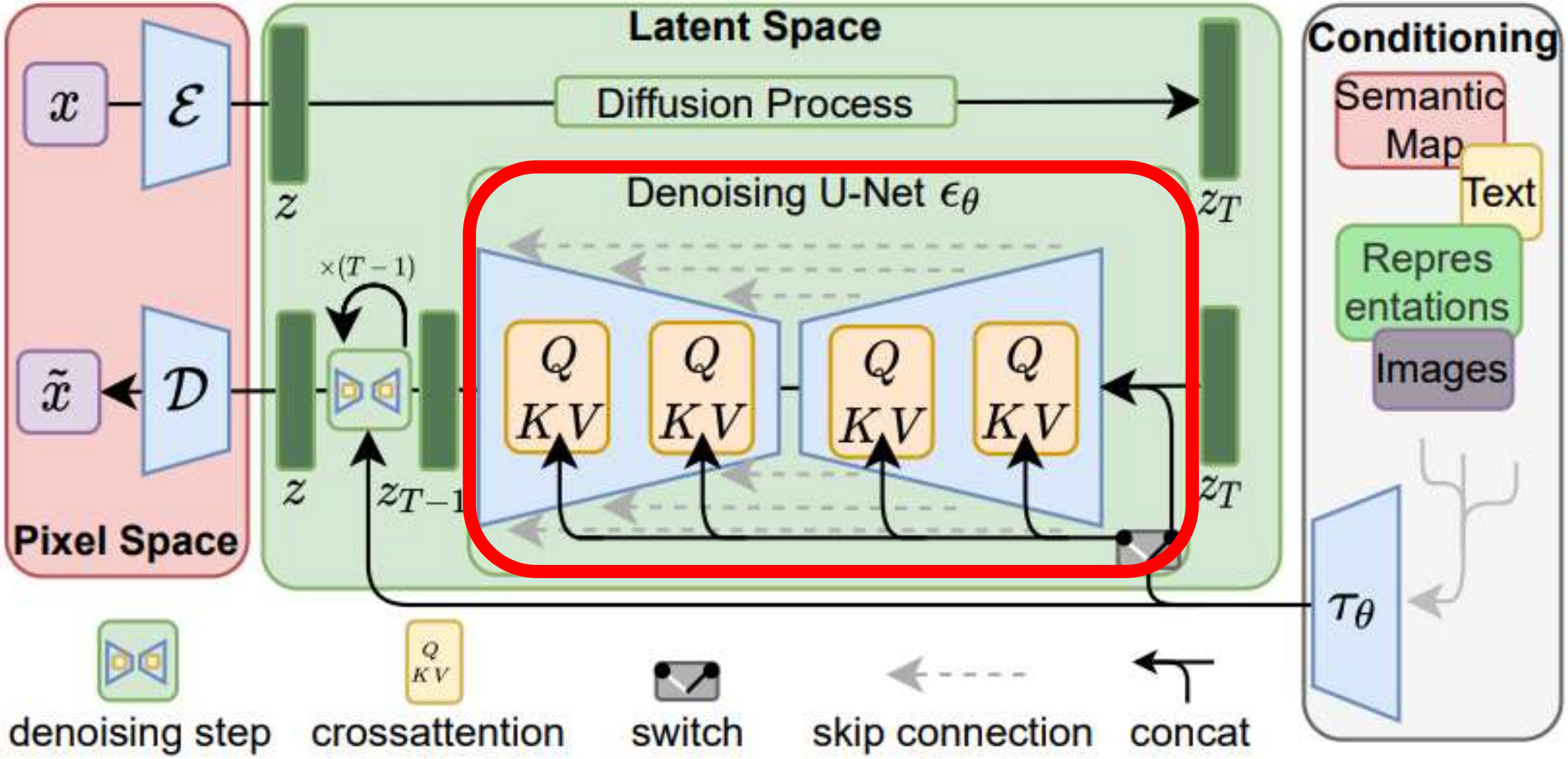
Label Conditional Diffusion

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

$$p(x_t \mid x_{t+1}, y) = \frac{p(x_t \mid x_{t+1}) p(y \mid x_t, x_{t+1})}{p(y \mid x_{t+1})} = \frac{\boxed{p(x_t \mid x_{t+1})} p(y \mid x_t)}{p(y \mid x_{t+1})},$$

Data Conditional Diffusion



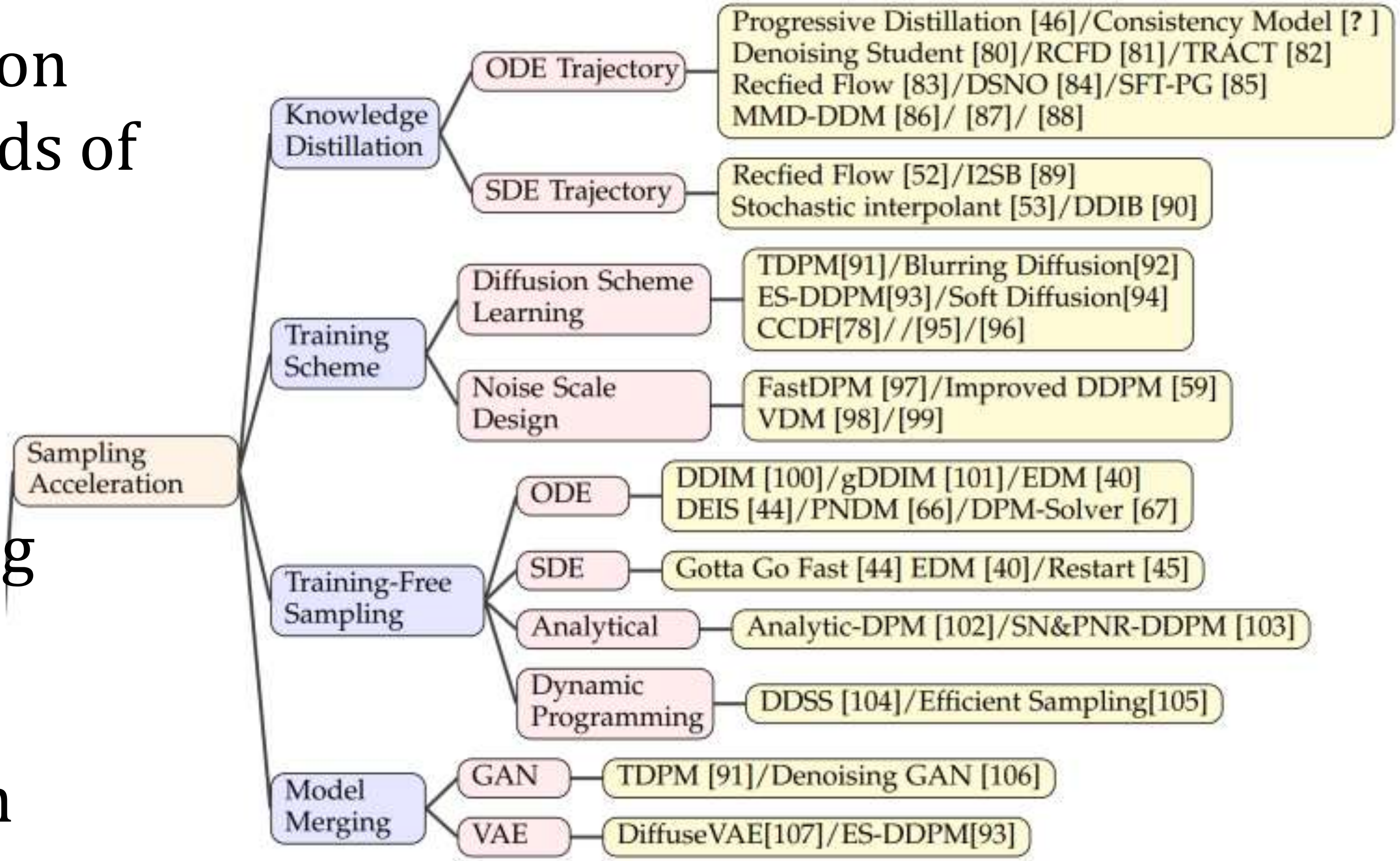
Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithms
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

Diffusion Needs Improvements

- Gaussian-based diffusion sampler takes thousands of steps to sample.

1. Training Scheme
2. Training-Free Sampling
3. Model Merging
4. Knowledge Distillation



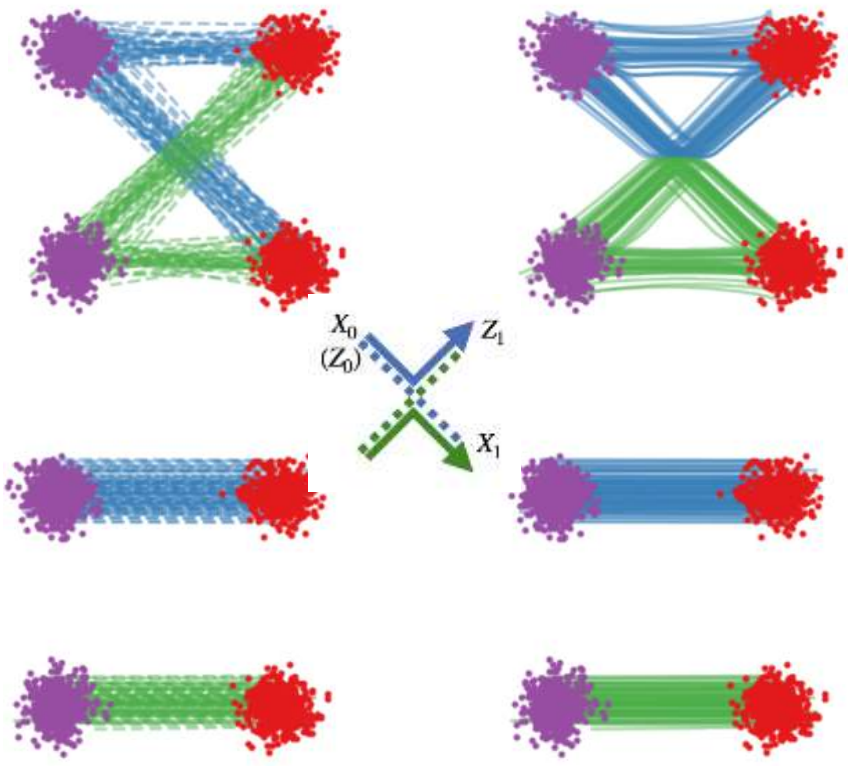
Diffusion Needs Improvements

1. Slow sampling
2. High-Dimensional Space
3. Conditional sampling
4. Wide range data application

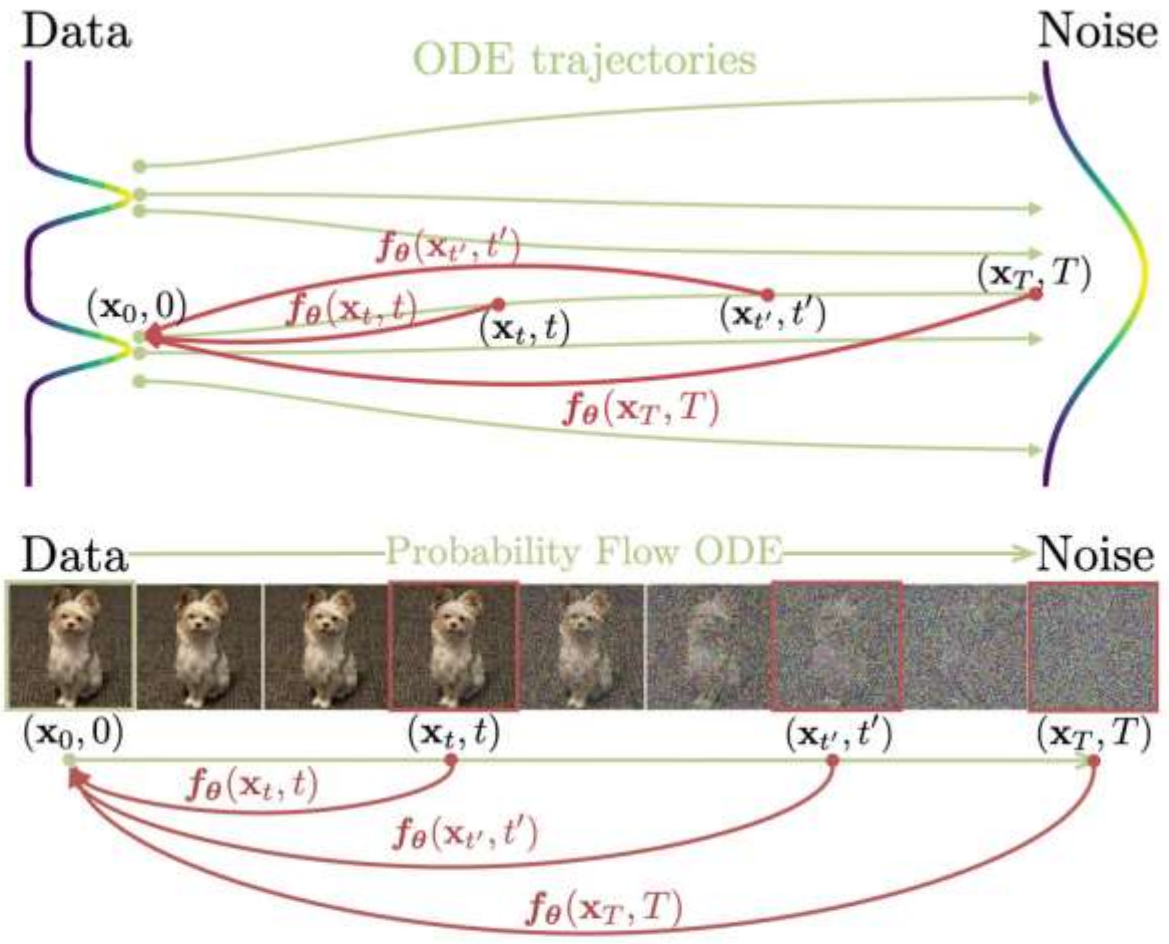
- Sampling Acceleration
- New Forward Process
- Likelihood Optimization
- Bridging Diffusion

Training-Free Sampling: Distillation

Path-based distillation



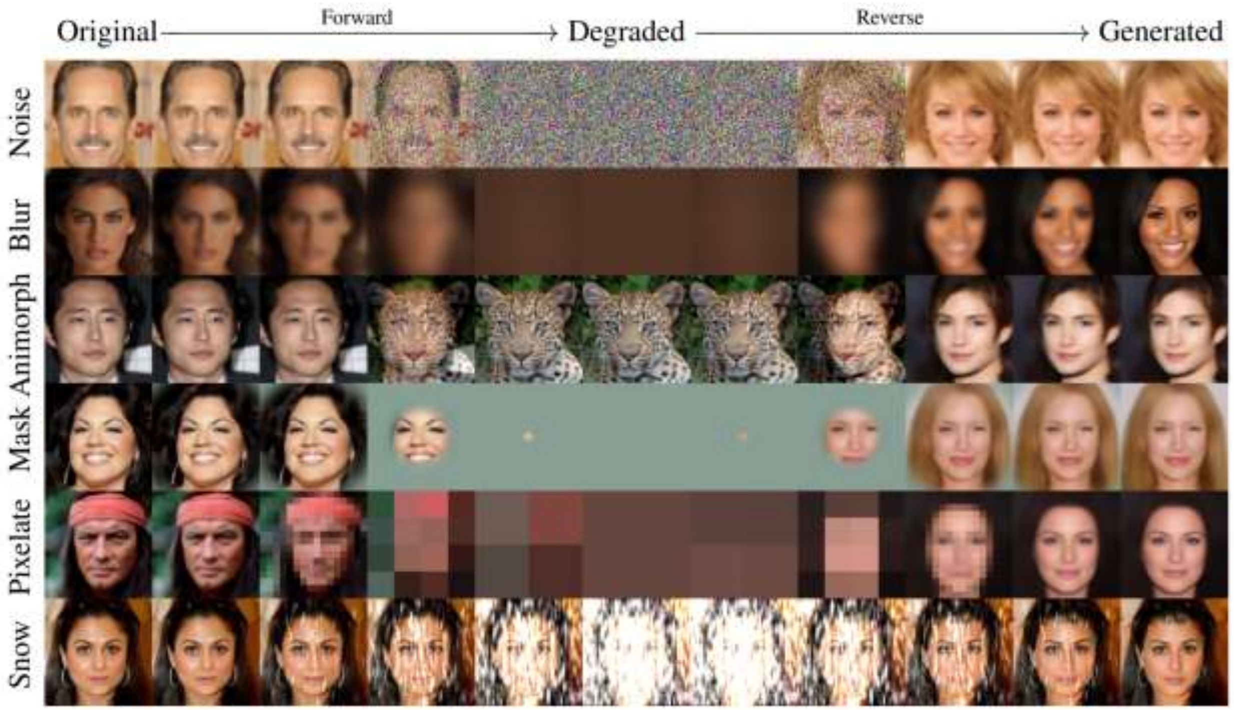
Objective-based distillation



Training Scheme: Diffusion Scheme Learning

Incomplete forward and sampling process → Non-Gaussian noise but a starting distribution from other distribution

Forward Path

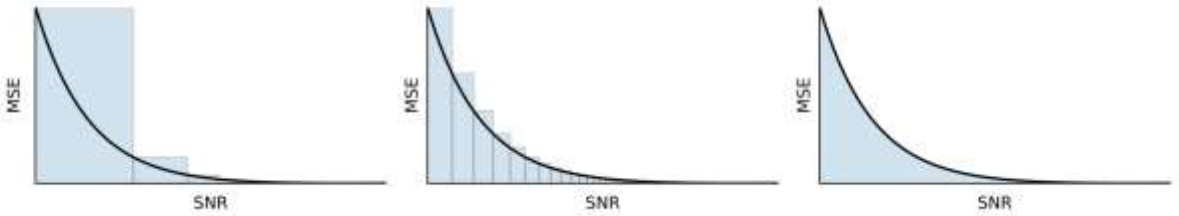


Efficient Noise

$$\mathrm{SNR(t)} := \alpha_t^2/\beta_t^2 = exp(\gamma_\eta(t)), \quad \sigma_t^2 = \mathrm{sigmoid}\left(\gamma_\eta(t)\right) \quad (16)$$

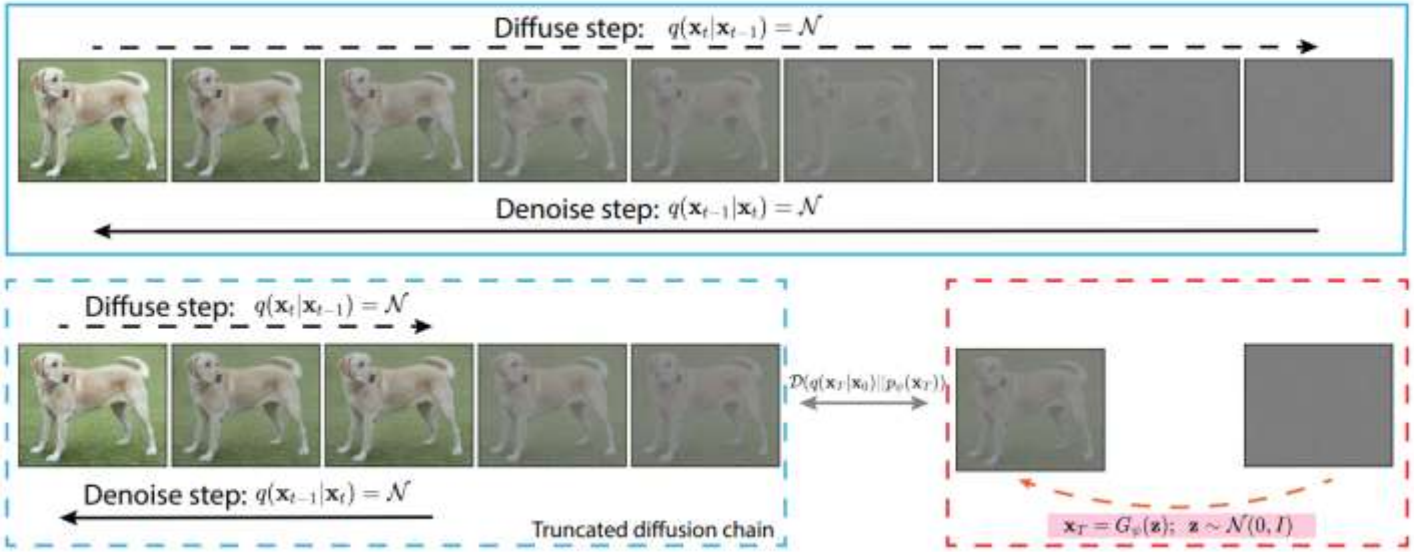
$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2}\mathbb{E}_{\epsilon\sim N(0,\mathbf{I})}\left[(\mathrm{SNR}(s) - \mathrm{SNR}(t))\left\|\mathbf{x} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t; t)\right\|_2^2\right]$$

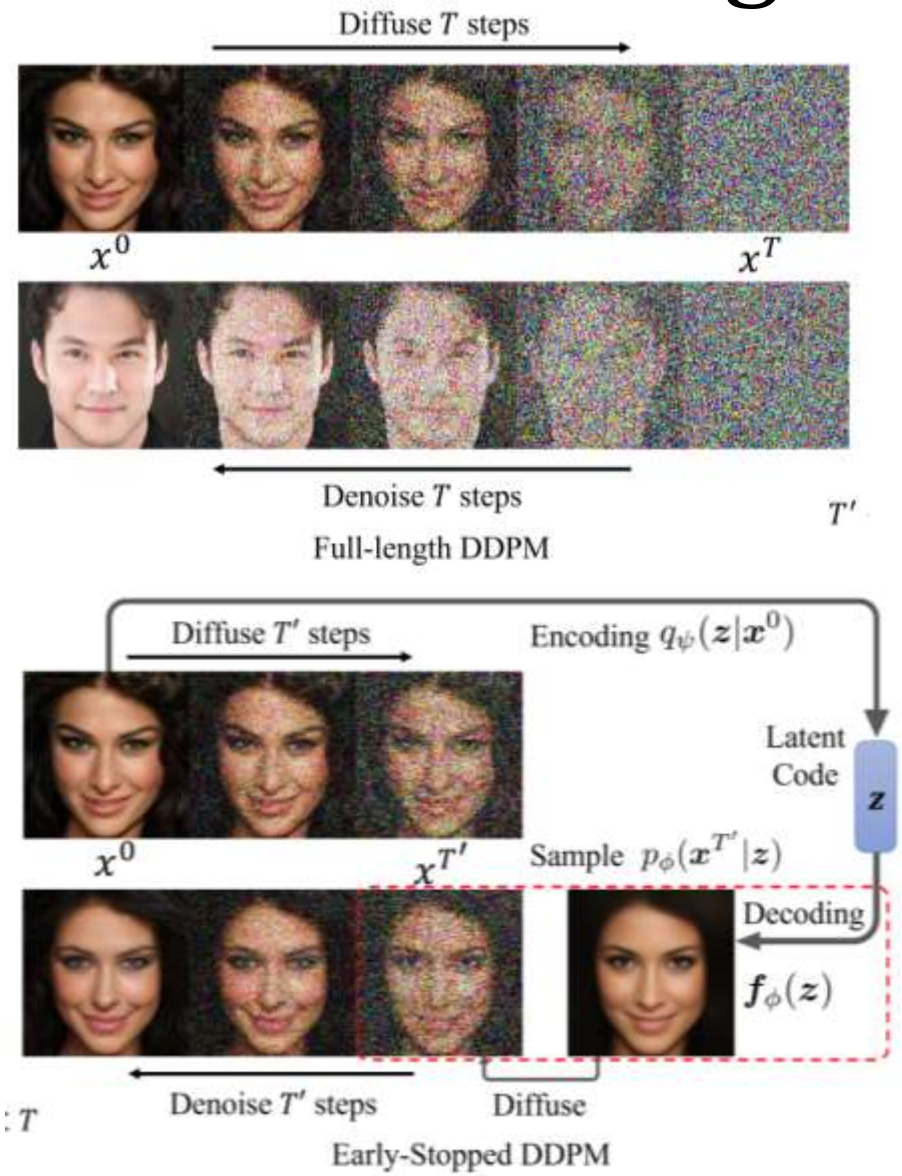
$$\mathcal{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\epsilon\sim N(0,\mathbf{I})}\int_{\mathrm{SNR}_{\min}}^{\mathrm{SNR}_{\max}}\left\|\mathbf{x} - \tilde{\mathbf{x}}_\theta(\mathbf{z}_v, v)\right\|_2^2 dv \quad (17)$$



Training Scheme: Diffusion Scheme Learning

Incomplete forward and sampling process → From intermediate states generated by other fast generative models (such as GAN and VAE)





Training Scheme: Noise Scale Design

- **FastDPM: link noise with time t with bijective map**

$$\mathcal{R}(t) = (\Delta\beta)^{\frac{t}{2}}\Gamma\left(\hat{\beta}+1\right)^{\frac{1}{2}}\Gamma\left(\hat{\beta}-t+1\right)^{-\frac{1}{2}}.$$

- **VDM: link noise with time t with bijective map**

$$\sigma_t^2 = \text{sigmoid}(\gamma_{\boldsymbol{\eta}}(t)) \qquad \alpha_t^2 = \text{sigmoid}(-\gamma_{\boldsymbol{\eta}}(t))$$

guide the training with SNR: $\text{SNR}(t) = \exp(-\gamma_{\boldsymbol{\eta}}(t))$

$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),i\sim U\{1,T\}}\left[(\text{SNR}(s)-\text{SNR}(t))\,||\mathbf{x}-\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)||_2^2\right],$$

$$\mathcal{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\int_{\text{SNR}_{\text{min}}}^{\text{SNR}_{\text{max}}}\|\mathbf{x}-\tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_v,v)\|_2^2\,dv,$$

- **Improved DDPM: learn noise scale by new loss function** $L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}}$

Algorithm 2 Sampling

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

Training-Free Sampling: DE In General

- Differential equation based samplers are actually numerical solvers for differential equations. Network's output is actually the step gradient for each steps
- The number of sampling steps depends on the errors during sampling

  -> Euler Method: $y_{n+1} = y_n + hf(t_n + y_n)$

  -> h: step size

  -> $f(t_n + y_n)$: estimated gradient
- Advanced sampler means advanced numerical solvers

  -> higher-order solvers: more accurate gradient -> larger step -> faster sampling

  -> multi-step solvers: multi-step results -> accurate predictions -> faster sampling

Training-Free Sampling: Other Techniques

Analytic Solvers

- Analytical DPM: Start from vlb optimization to explore analytical solutions

- Extended Analytical DPM: Suppose that reverse mean is independent of reverse noise, finding optimal mean and noise

Theorem 1. *(Score representation of the optimal solution to Eq. (4), proof in Appendix A.2)*

The optimal solution $\boldsymbol{\mu}_n^*(\boldsymbol{x}_n)$ and σ_n^{*2} to Eq. (4) are

$$\boldsymbol{\mu}_n^*(\boldsymbol{x}_n) = \tilde{\boldsymbol{\mu}}_n\left(\boldsymbol{x}_n, \frac{1}{\sqrt{\alpha_n}}(\boldsymbol{x}_n + \overline{\beta}_n \nabla_{\boldsymbol{x}_n} \log q_n(\boldsymbol{x}_n))\right),$$

$$\sigma_n^{*2} = \lambda_n^2 + \left(\sqrt{\frac{\overline{\beta}_n}{\alpha_n}} - \sqrt{\overline{\beta}_{n-1} - \lambda_n^2}\right)^2 \left(1 - \overline{\beta}_n \mathbb{E}_{q_n(\boldsymbol{x}_n)} \frac{||\nabla_{\boldsymbol{x}_n} \log q_n(\boldsymbol{x}_n)||^2}{d}\right).$$

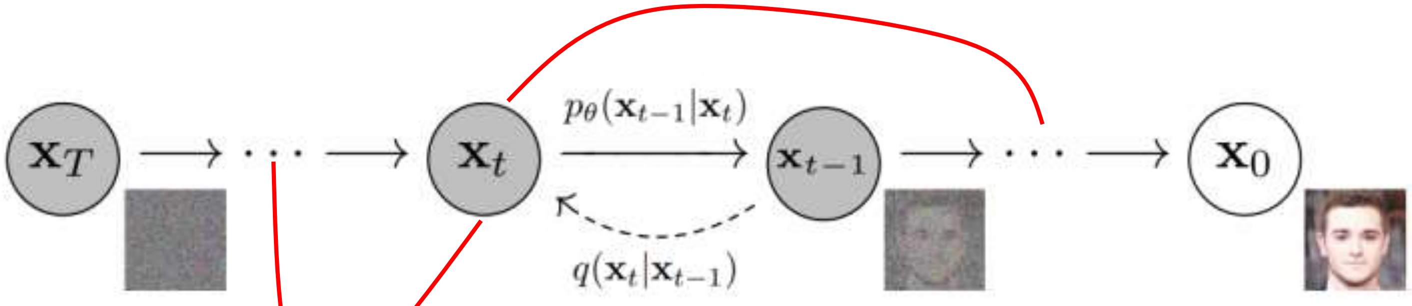
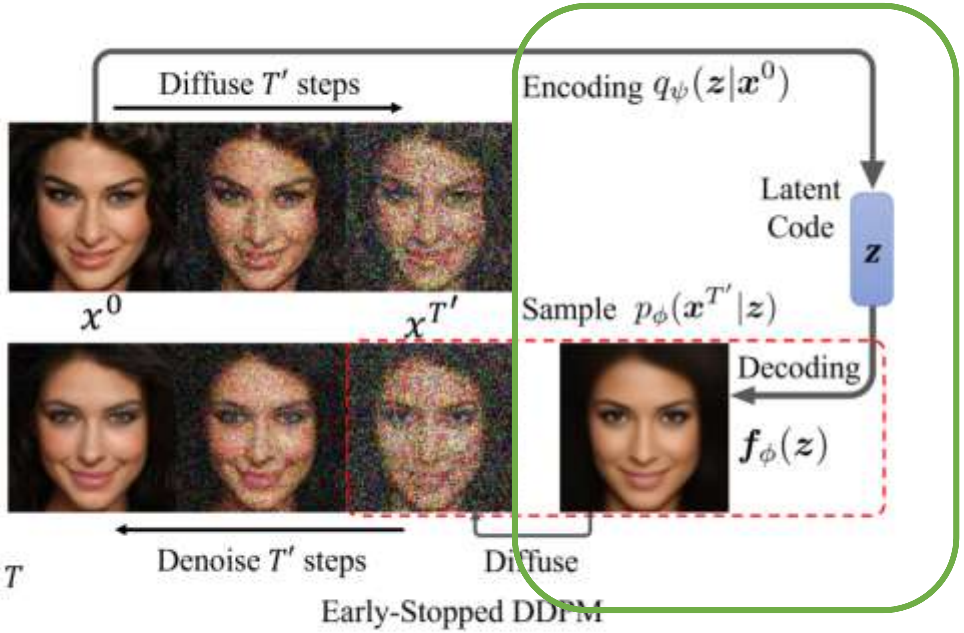


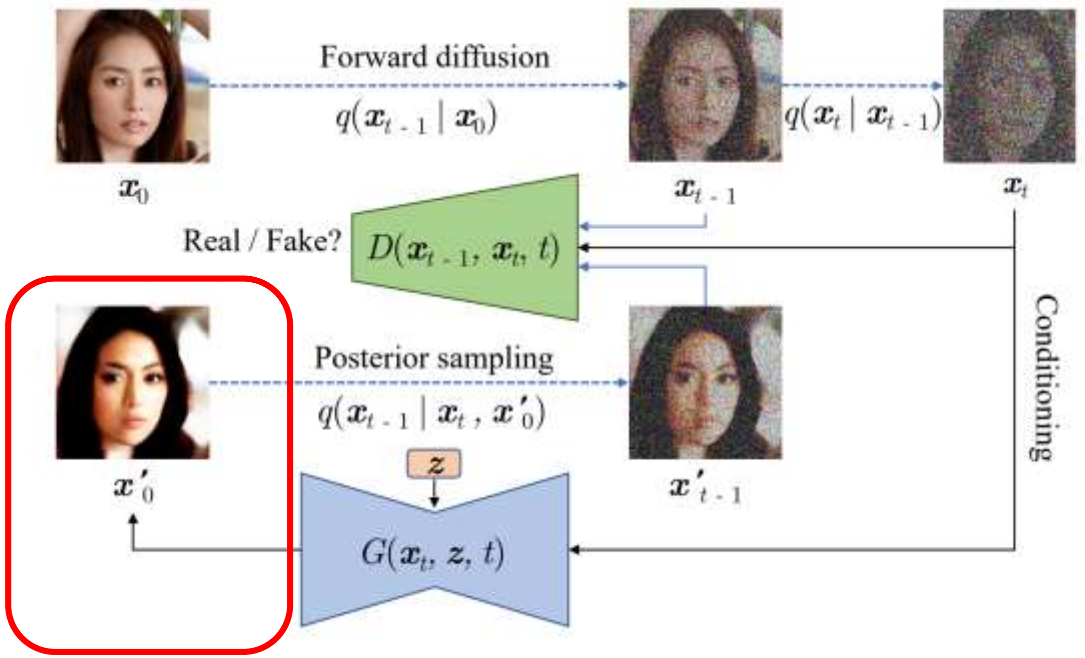
Figure 2: The directed graphical model considered in this work.

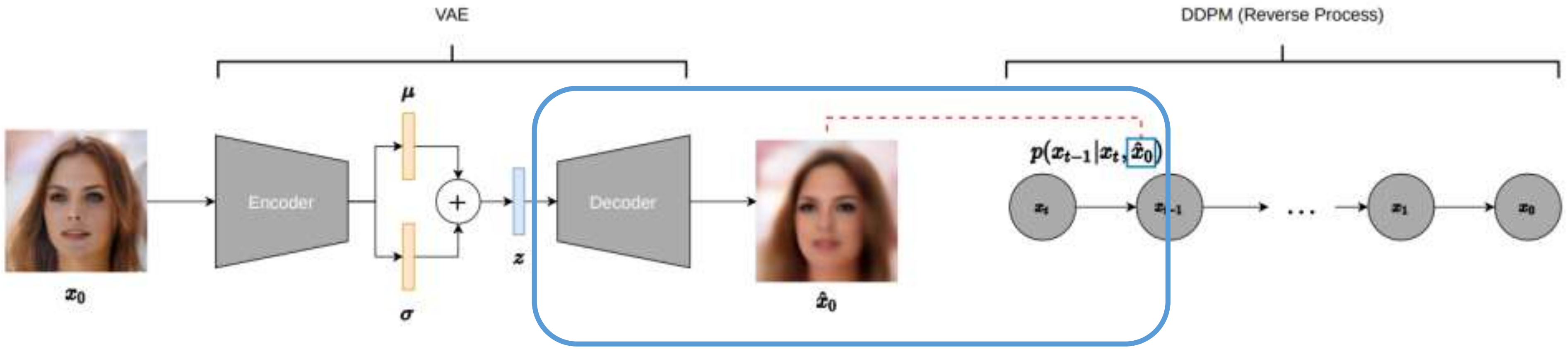
Dynamic Programming

- Construct a refinement path composed of K sampling steps according to the log-likelihood losses.

Model Merging: Acceleration

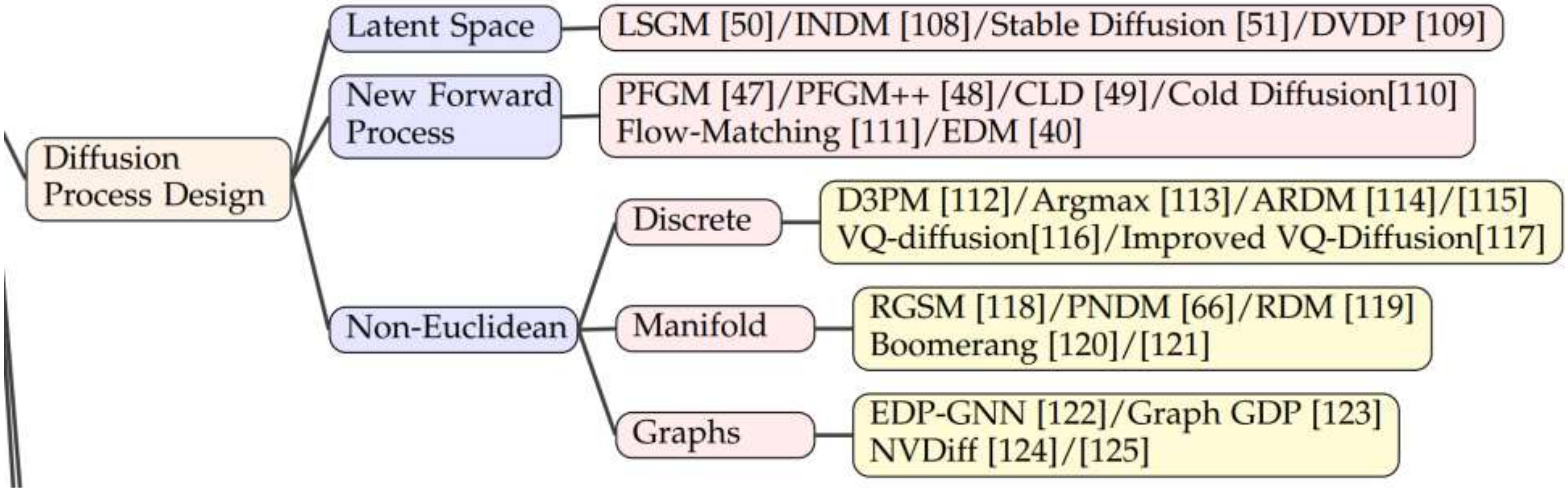




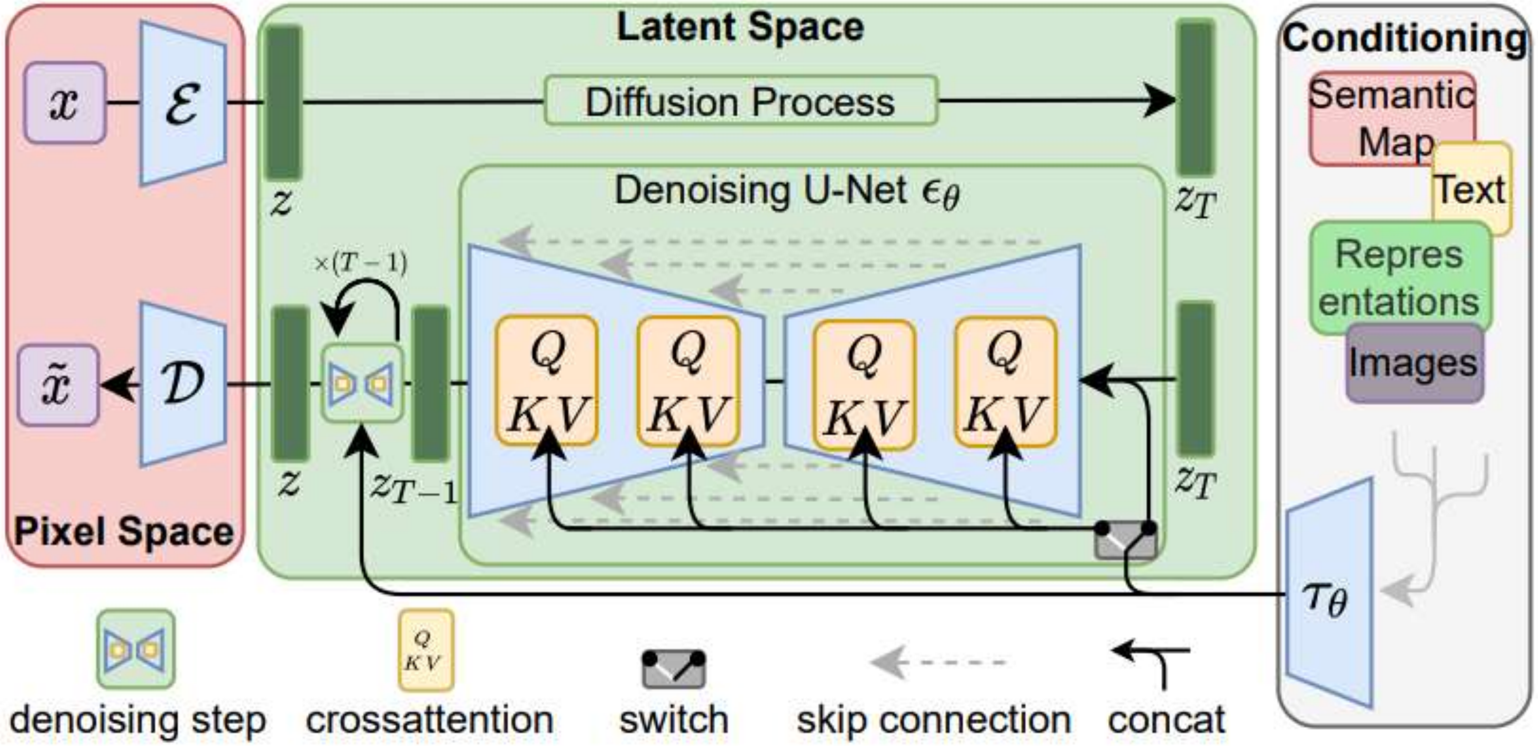


Diffusion Needs Improvements

Designing advanced process for wider application, including:
diverse data types, cross-modality generation, cross distribution



Diffusion Process Design: Latent Space

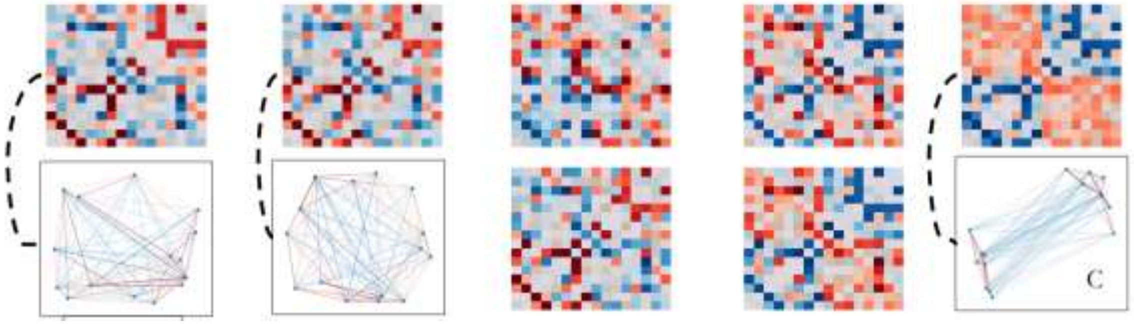


Latent Diffusion: Conducting diffusion process on latent space

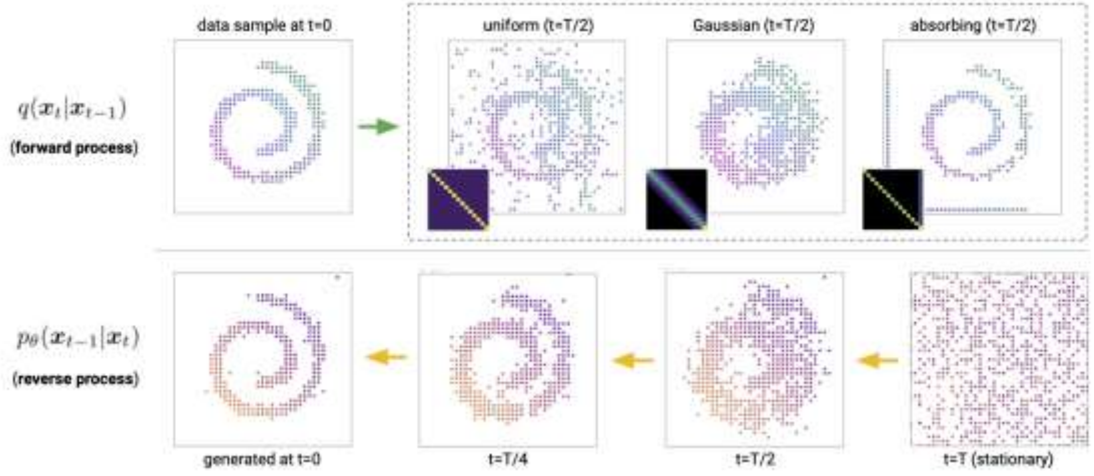
- Faster convergence
- Mutli-condition guidance
- Prior knowledge from Encoders & Decoders

Diffusion Process Design: Wider Range

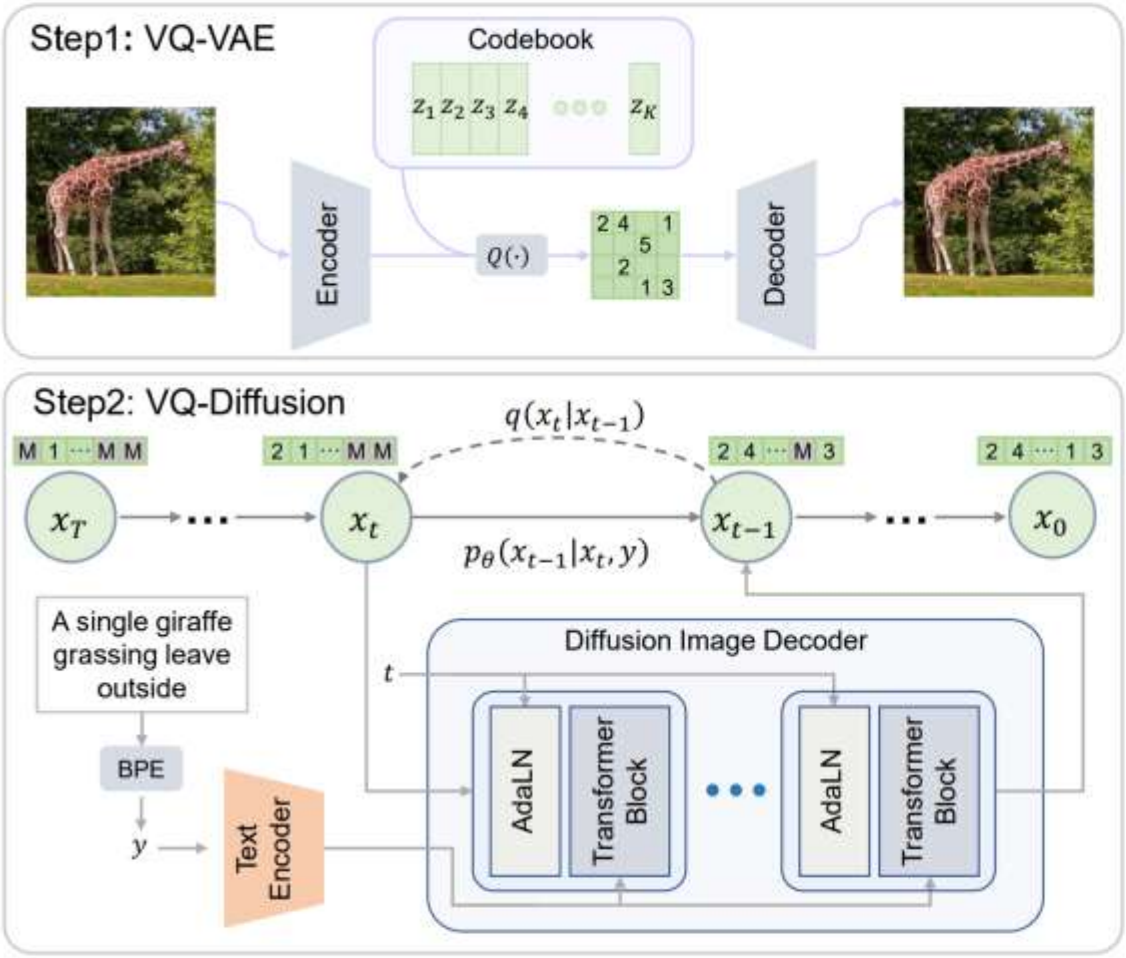
Graph



Categorical data

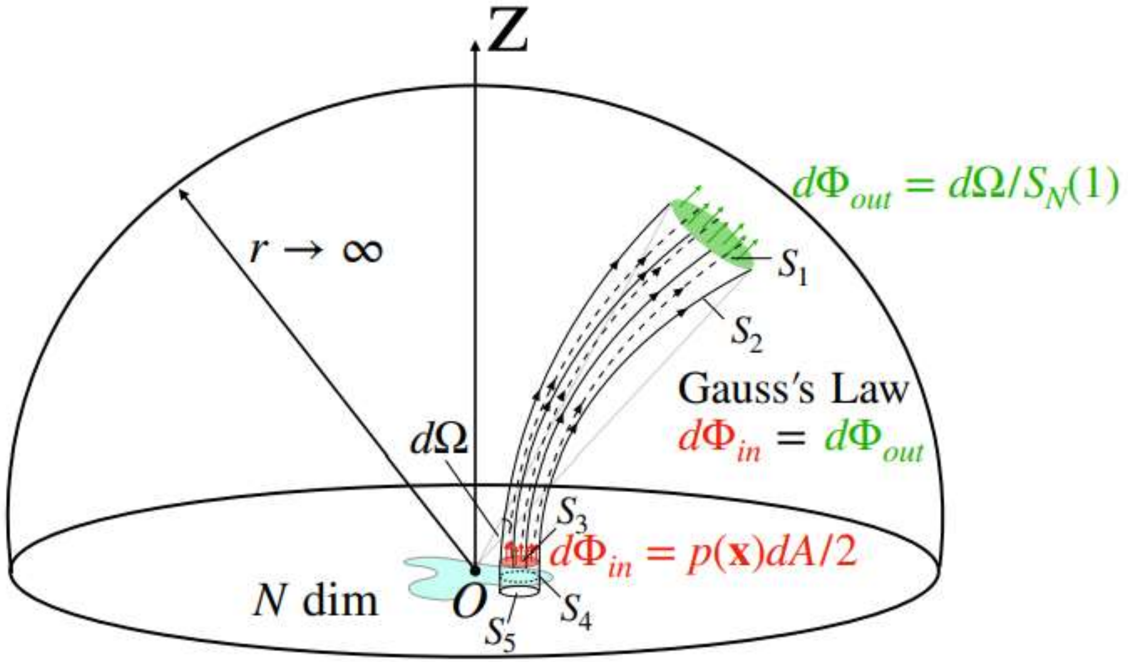


Vector-Quantized: Cross modality

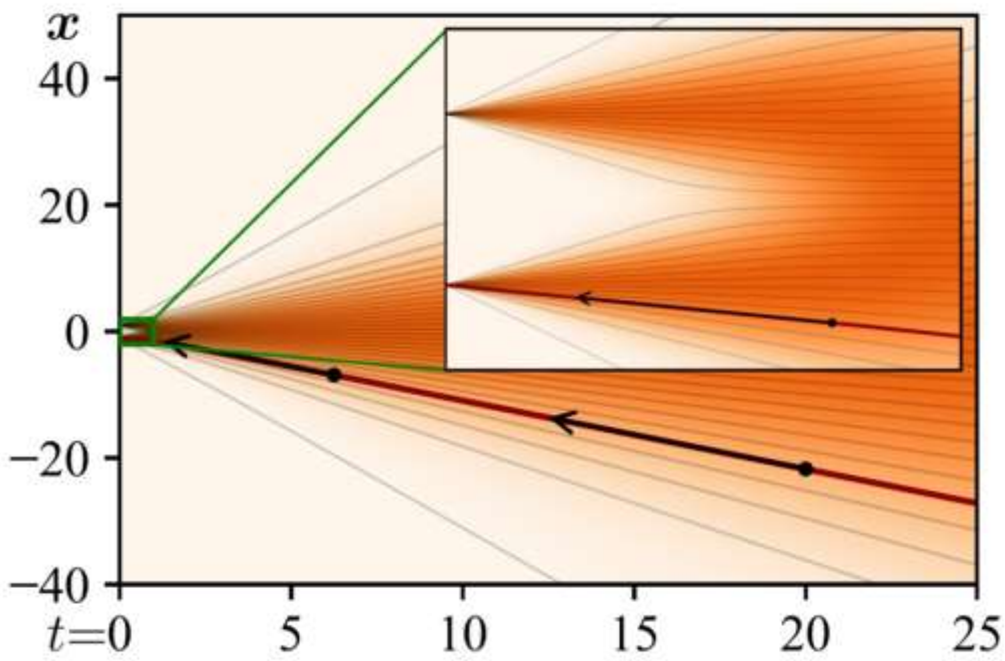


Diffusion Process Design: Advanced Forward

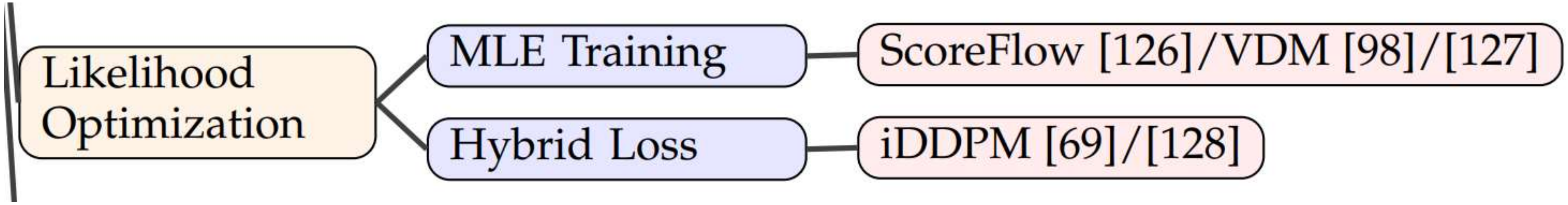
Physics-inspired forward process



Self-designed sampling space



Diffusion Needs Improvements



Enhance the continuous diffusion training from the perspective of

- Directly optimizing the likelihood instead of
- Minimizing the lower bound of log-likelihood

Likelihood Optimization: Advanced ELBO

Score Connection:

Represent ELBO based on score-matching loss

Theorem 3. *Let $p_{0t}(\mathbf{x}' \mid \mathbf{x})$ denote the transition distribution from $p_0(\mathbf{x})$ to $p_t(\mathbf{x})$ for the SDE in Eq. (1). With the same notations and conditions in Theorem 1, we have*

$$-\log p_{\boldsymbol{\theta}}^{\mathrm{SDE}}(\mathbf{x}) \leqslant \mathcal{L}_{\boldsymbol{\theta}}^{SM}(\mathbf{x}) = \mathcal{L}_{\boldsymbol{\theta}}^{DSM}(\mathbf{x}), \tag{11}$$

where $\mathcal{L}_{\boldsymbol{\theta}}^{SM}(\mathbf{x})$ is defined as

$$-\mathbb{E}_{p_{0T}(\mathbf{x}'|\mathbf{x})}[\log \pi(\mathbf{x}')] + \frac{1}{2}\int_0^T \mathbb{E}_{p_{0t}(\mathbf{x}'|\mathbf{x})}\left[2g(t)^2 \nabla_{\mathbf{x}'} \cdot \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}', t) + g(t)^2 \left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}', t)\right\|_2^2 - 2\nabla_{\mathbf{x}'} \cdot \boldsymbol{f}(\mathbf{x}', t)\right] \mathrm{d}t,$$

and $\mathcal{L}_{\boldsymbol{\theta}}^{DSM}(\mathbf{x})$ is given by

$$\begin{aligned} &-\mathbb{E}_{p_{0T}(\mathbf{x}'|\mathbf{x})}[\log \pi(\mathbf{x}')] + \frac{1}{2}\int_0^T \mathbb{E}_{p_{0t}(\mathbf{x}'|\mathbf{x})}\left[g(t)^2 \left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}', t) - \nabla_{\mathbf{x}'} \log p_{0t}(\mathbf{x}' \mid \mathbf{x})\right\|_2^2\right] \mathrm{d}t \\ &\qquad - \frac{1}{2}\int_0^T \mathbb{E}_{p_{0t}(\mathbf{x}'|\mathbf{x})}\left[g(t)^2 \left\|\nabla_{\mathbf{x}'} \log p_{0t}(\mathbf{x}' \mid \mathbf{x})\right\|_2^2 + 2\nabla_{\mathbf{x}'} \cdot \boldsymbol{f}(\mathbf{x}', t)\right] \mathrm{d}t. \end{aligned}$$

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}}$$

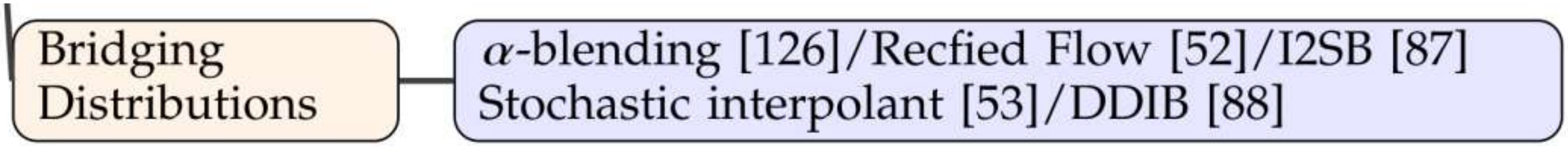
$$\mathcal{L}_T(\mathbf{x}) = \frac{T}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}), i\sim U\{1,T\}}\left[(\mathrm{SNR}(s) - \mathrm{SNR}(t))\left\|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\right\|_2^2\right],$$

$$\mathcal{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\int_{\mathrm{SNR}_{\mathrm{min}}}^{\mathrm{SNR}_{\mathrm{max}}} \left\|\mathbf{x} - \tilde{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{z}_v, v)\right\|_2^2 \, dv,$$

Re-Design:

Design vlb loss from a different perspective to obtain better convergence

Diffusion Needs Improvements



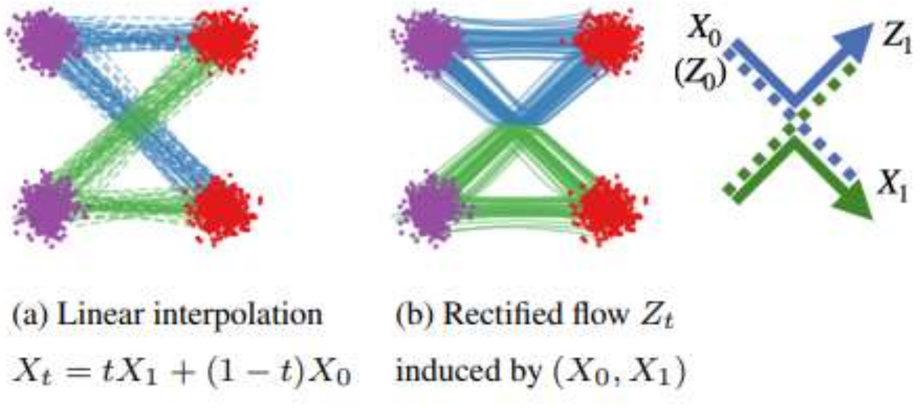
From one-direction translation to bi-directional translation:

- Image-to-image translation
- Cell distribution transportation

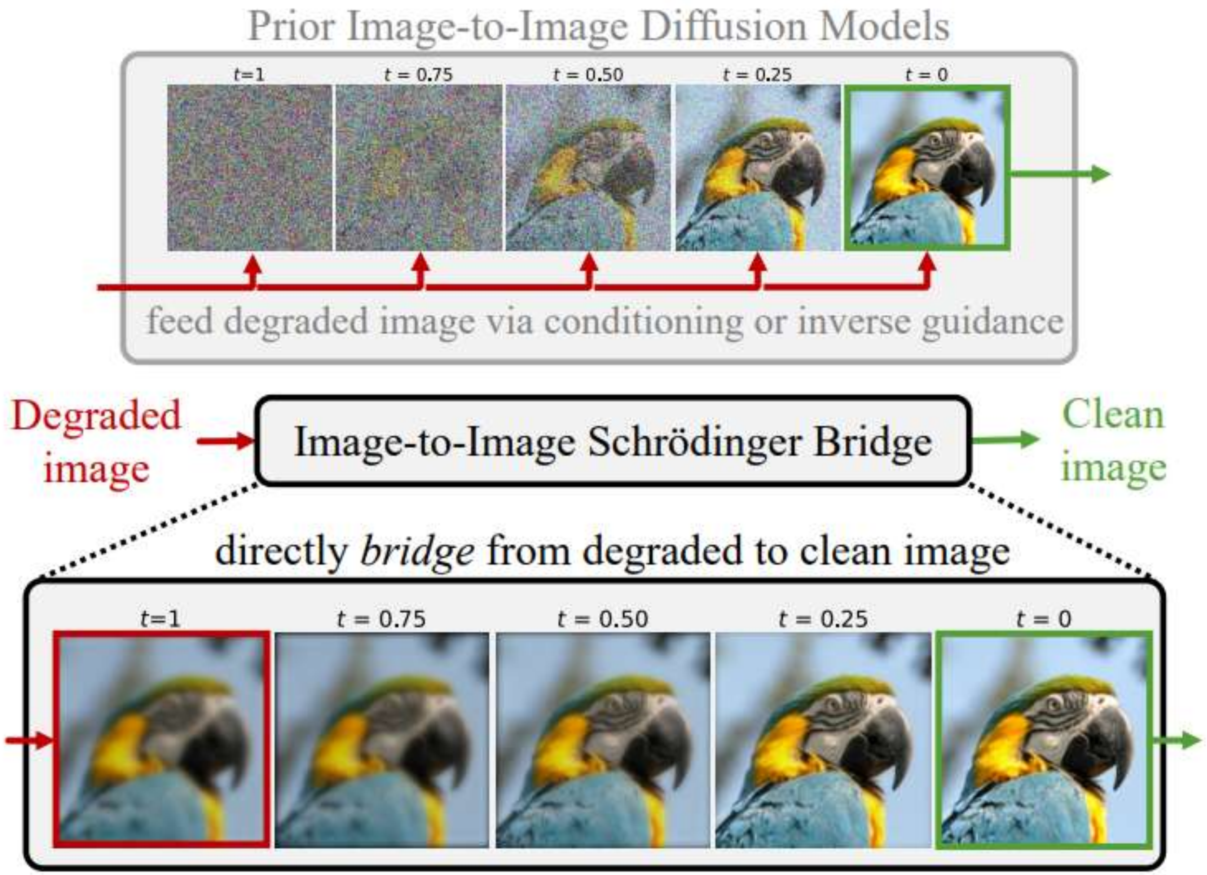
Bridging Distributions: Multi-directional

Connecting two distributions:

- Design transportation maps based on score-matching objective
- Apply Schrodinger bridge for the connection



Optimal Transport

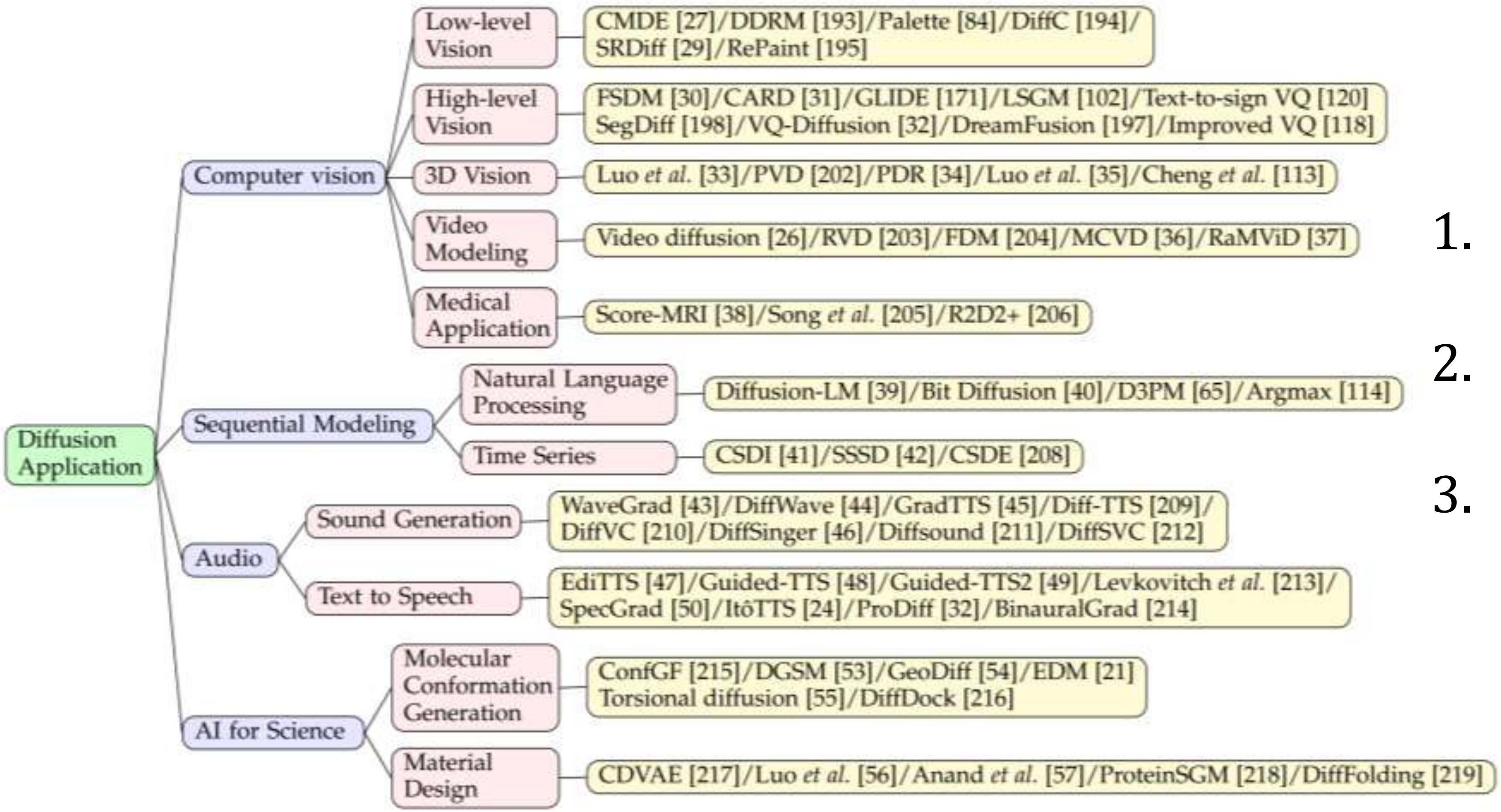


Schrodinger Bridge

Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithms
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

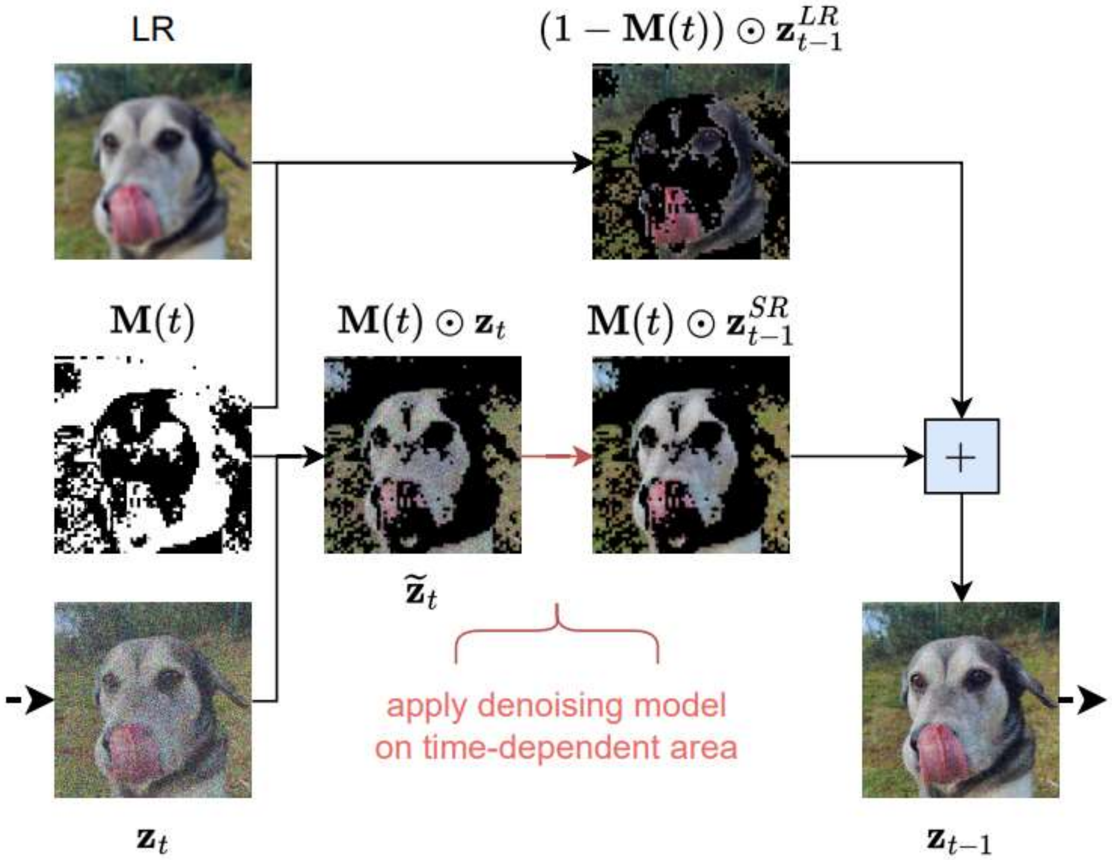
Diffusion Applications



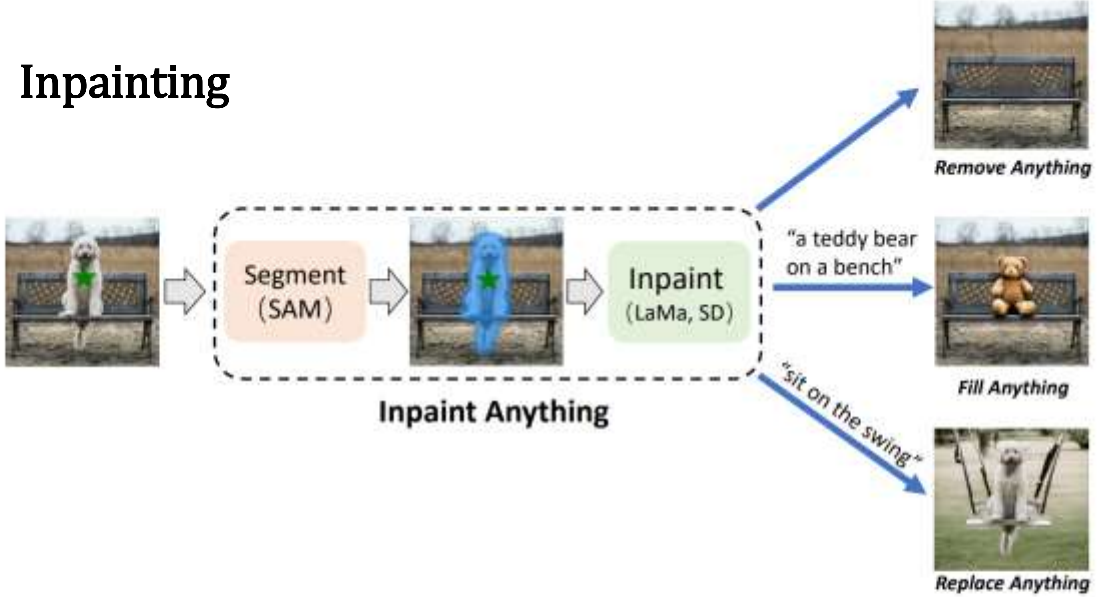
1. Mechanisms behind the better performance
2. Key-point technique in the implementation
3. Current challenges and future directions

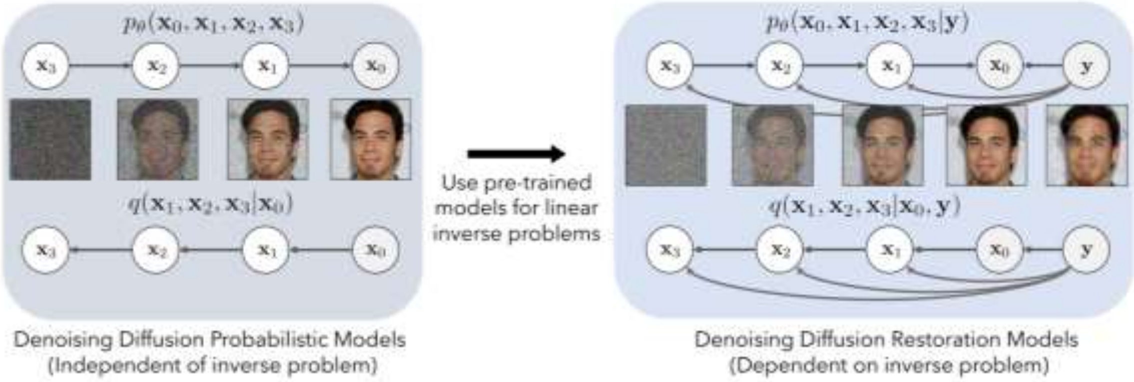
Low-Level Vision

Super-Resolution



Inpainting

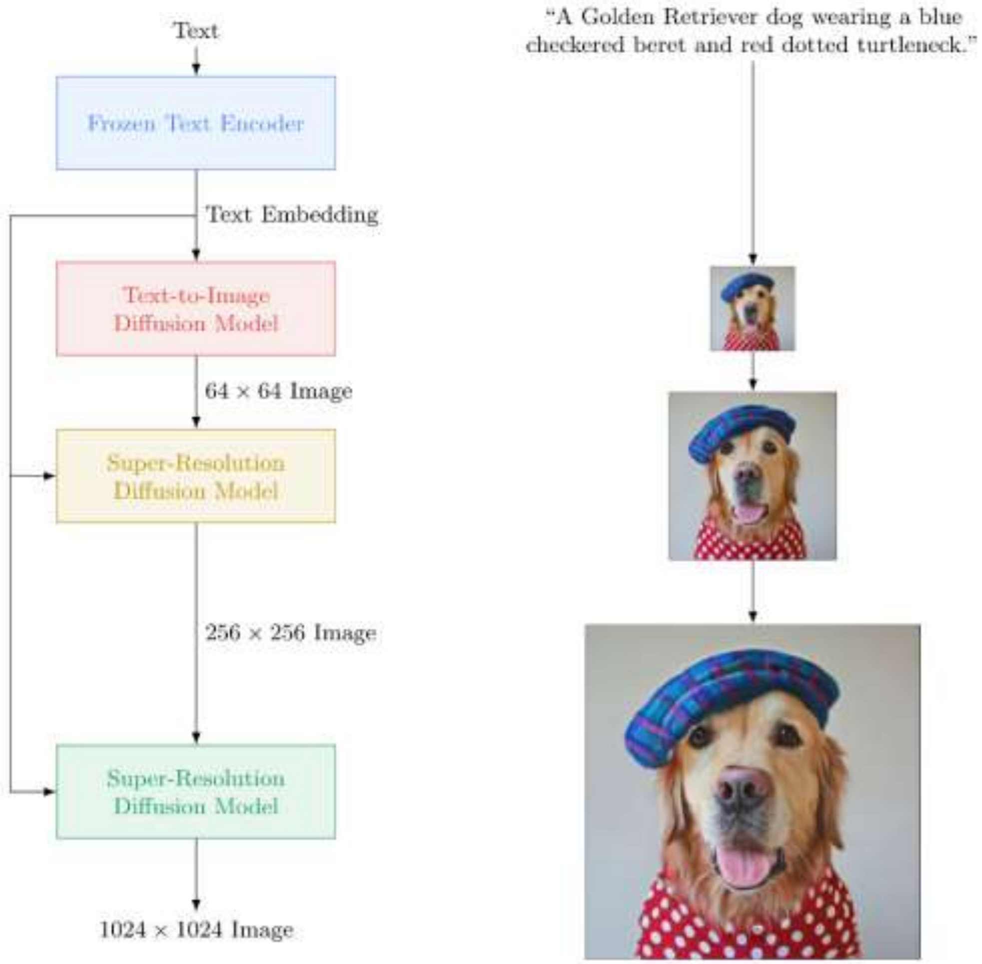




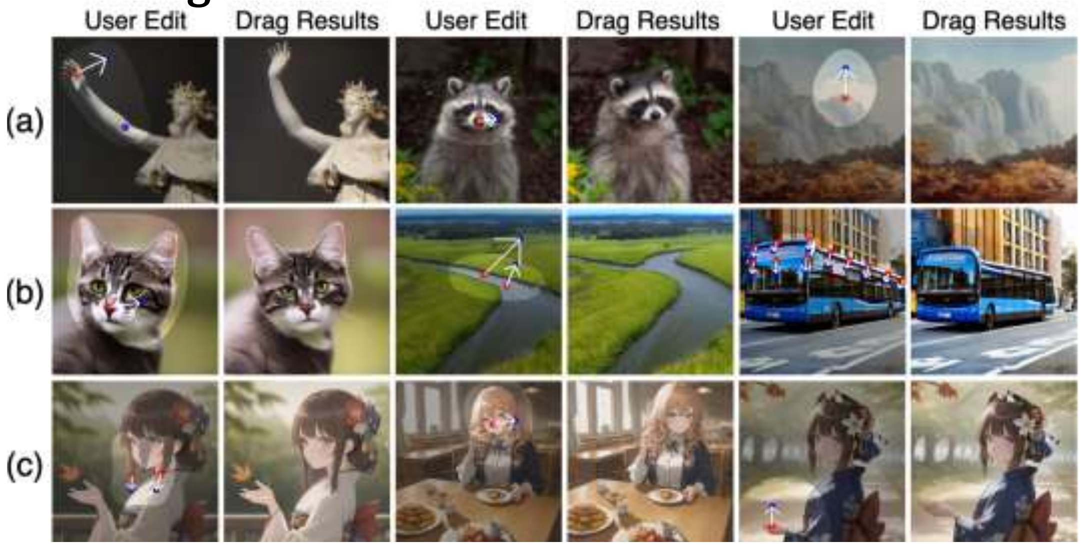
Restoration

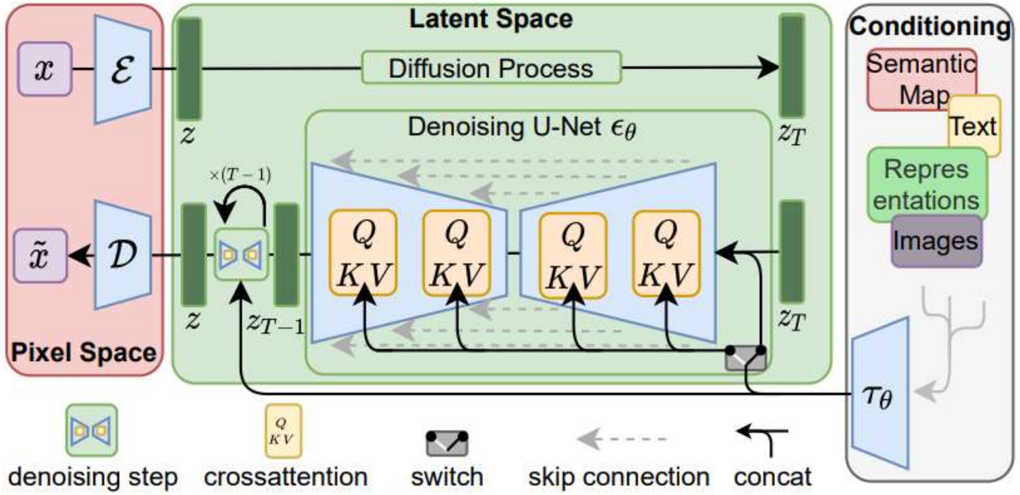
High-Level Vision

Generation



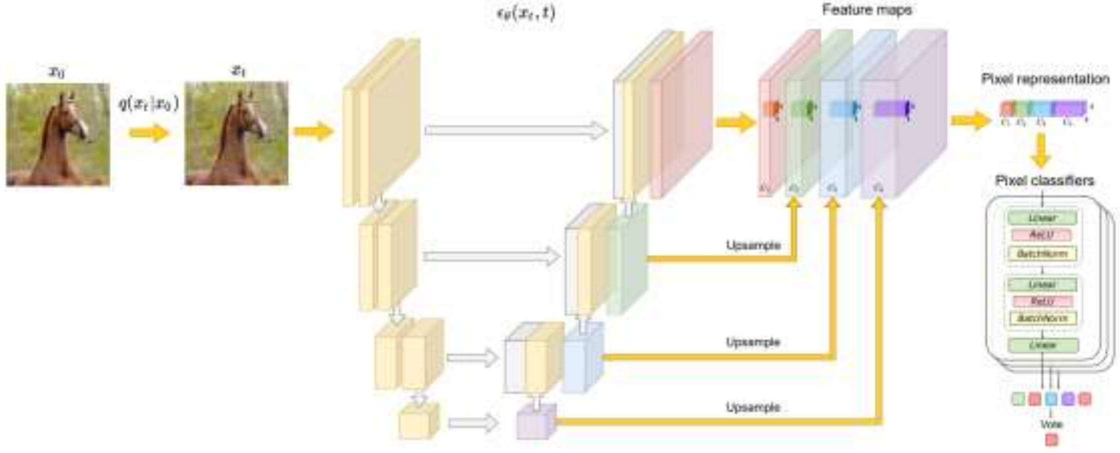
Editing

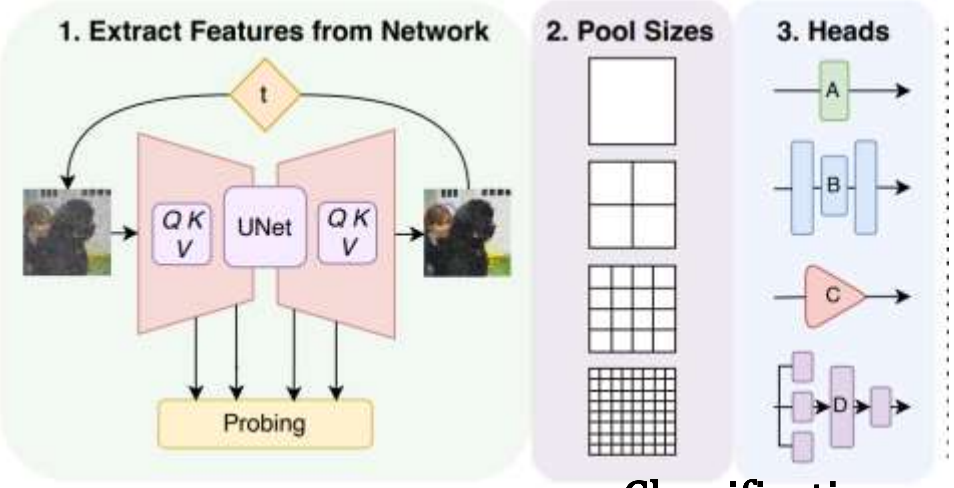




High-Level Vision

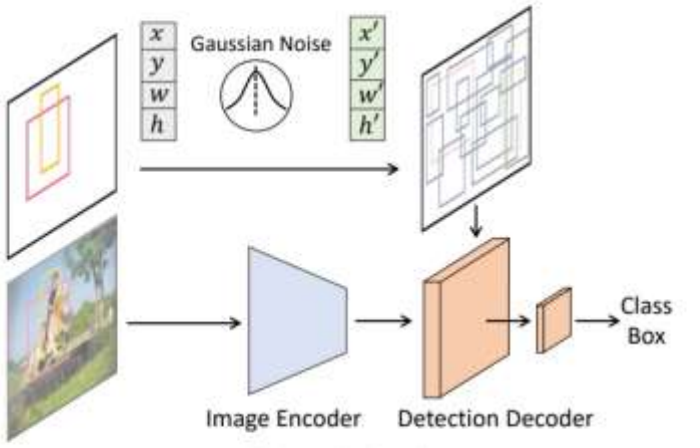
Segmentation



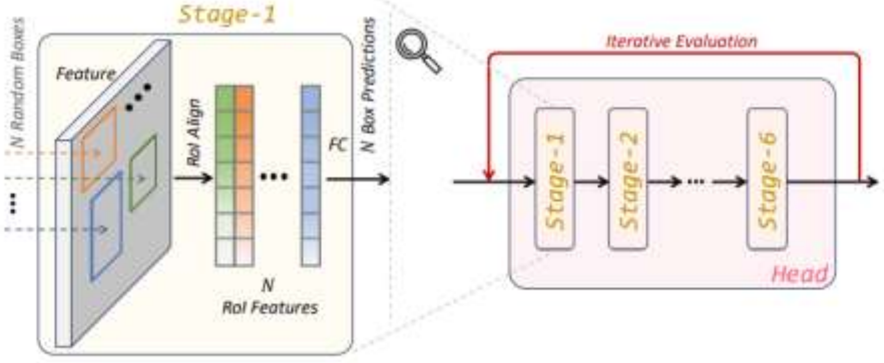


Classification

Detection



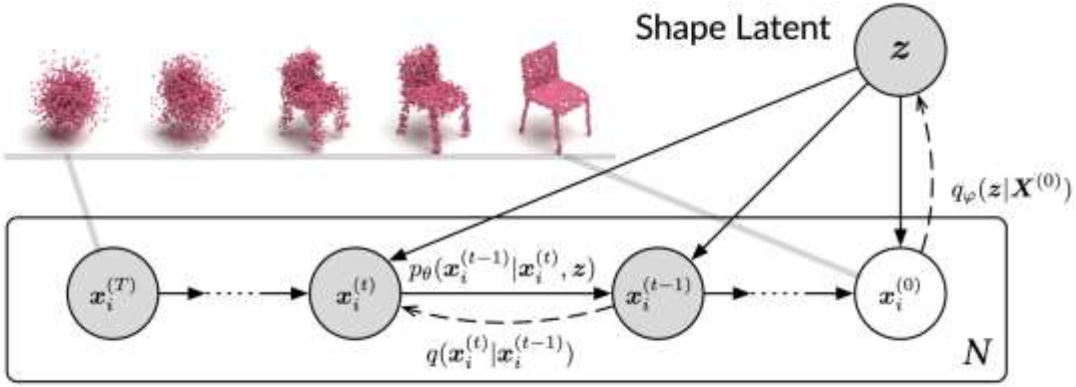
(a) Overall pipeline.

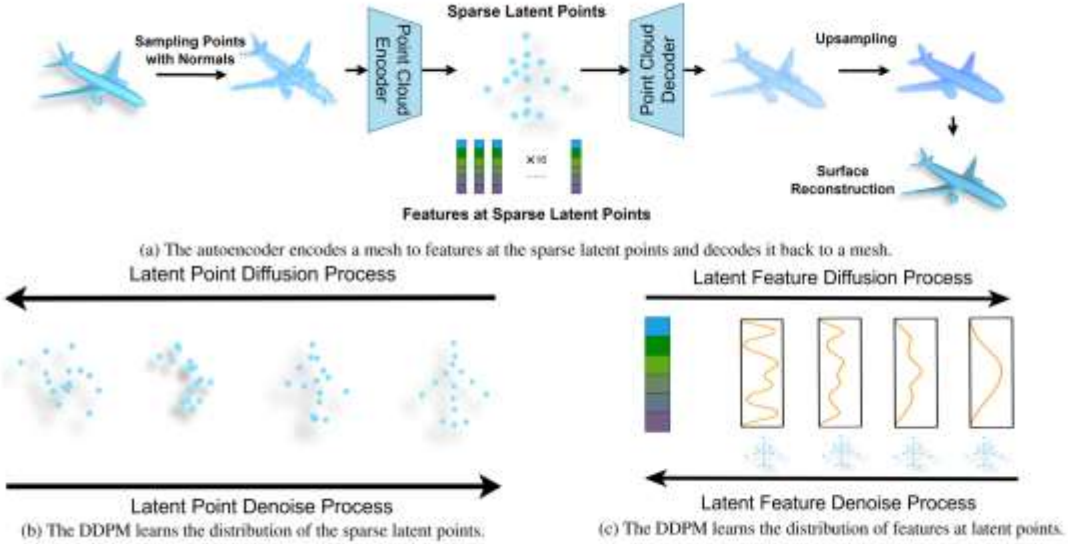


(b) Details of the detection decoder/head.

3D Vision

Unconditional Generation





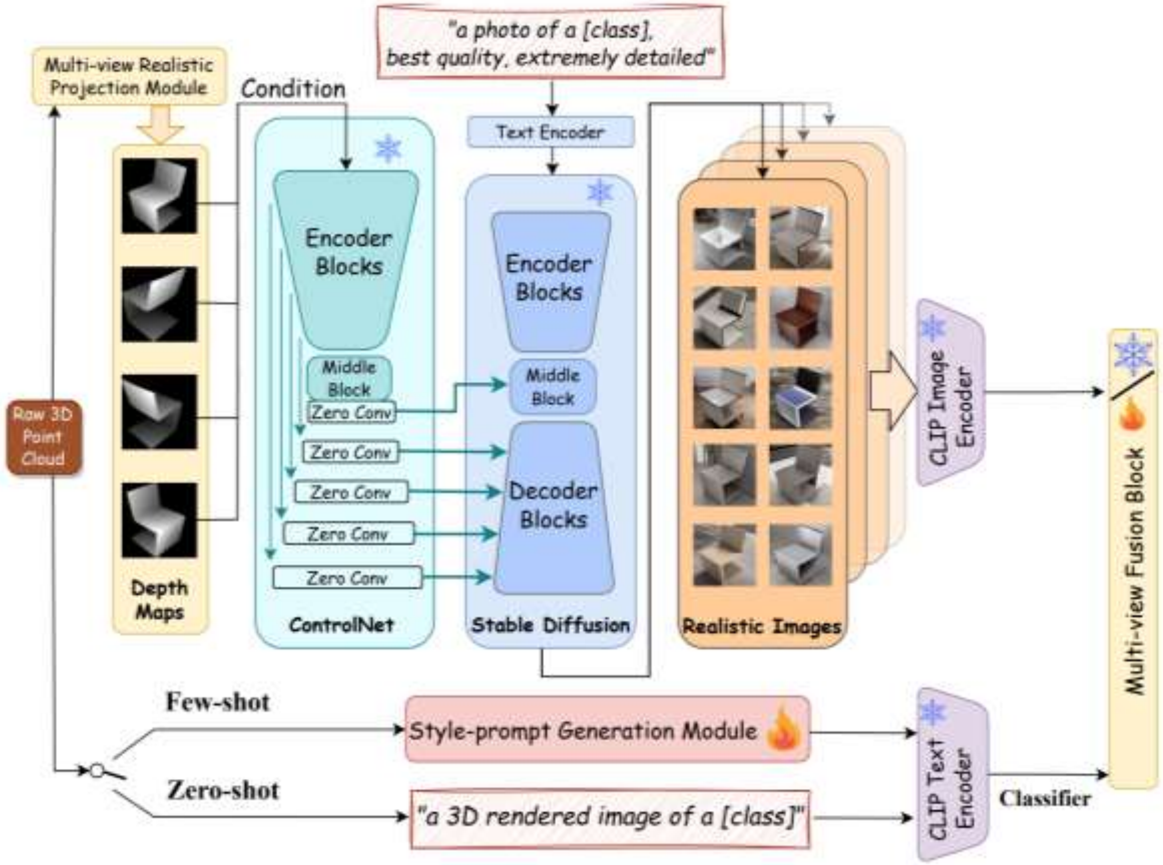
(a) The autoencoder encodes a mesh to features at the sparse latent points and decodes it back to a mesh.

(b) The DDPM learns the distribution of the sparse latent points.

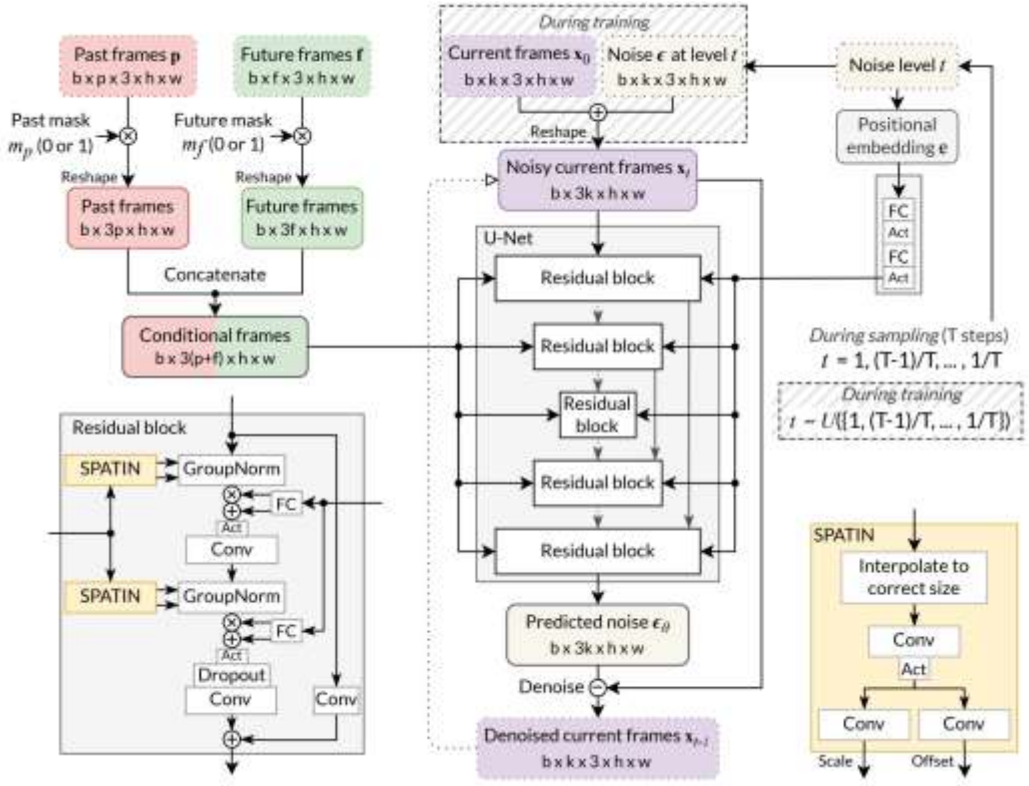
(c) The DDPM learns the distribution of features at latent points.

Completion

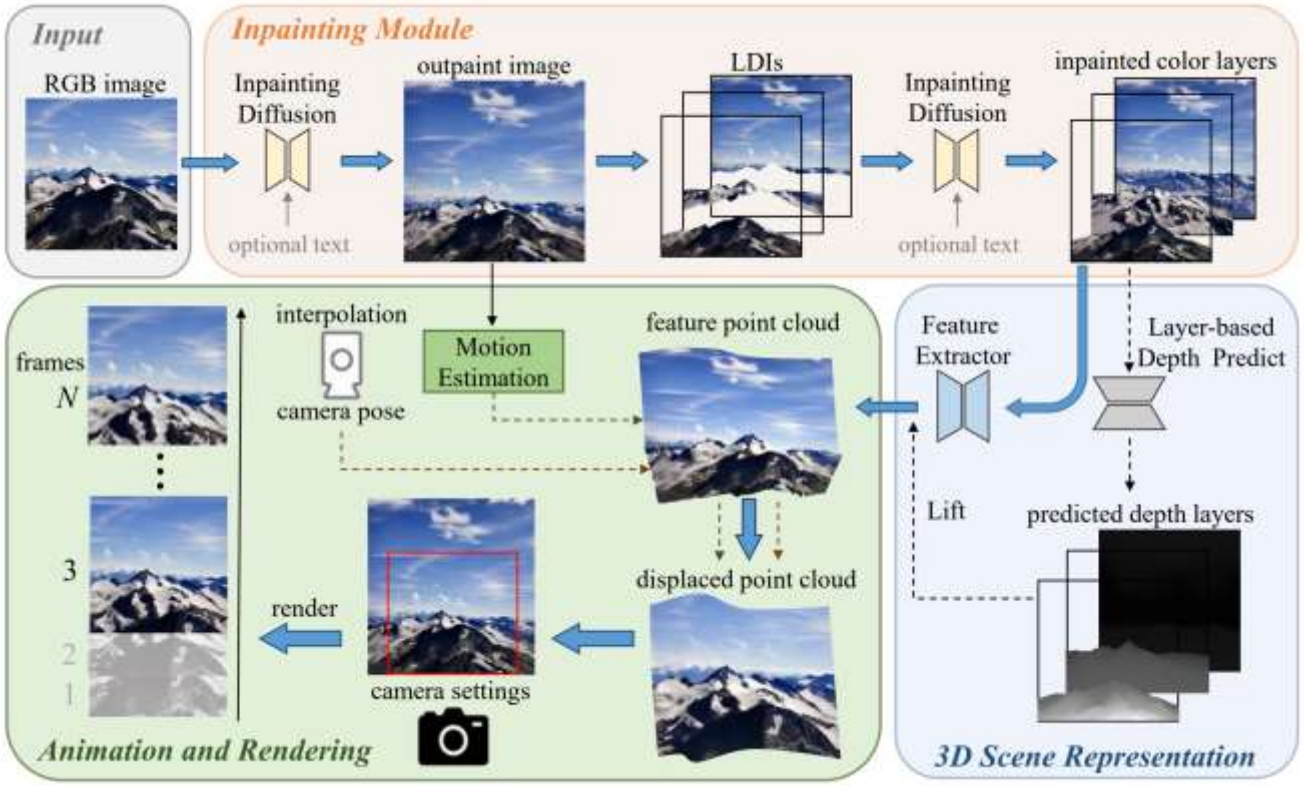
Multi-Conditional Generation



Video Modeling

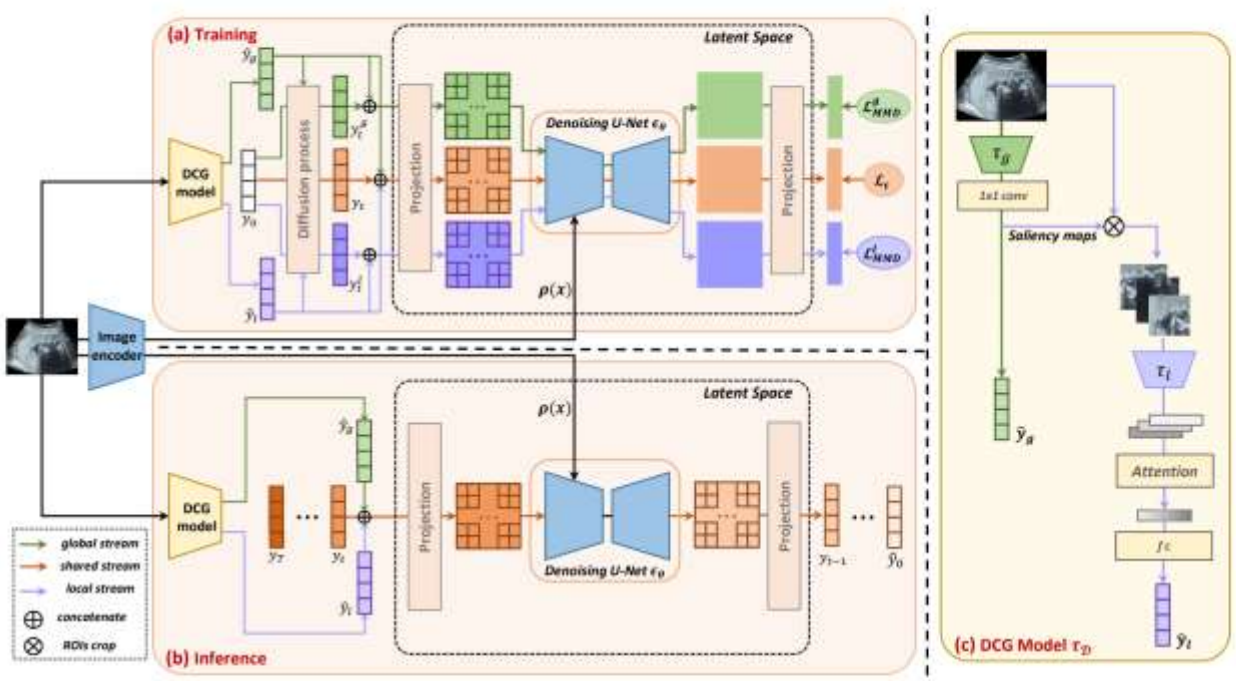


Generation



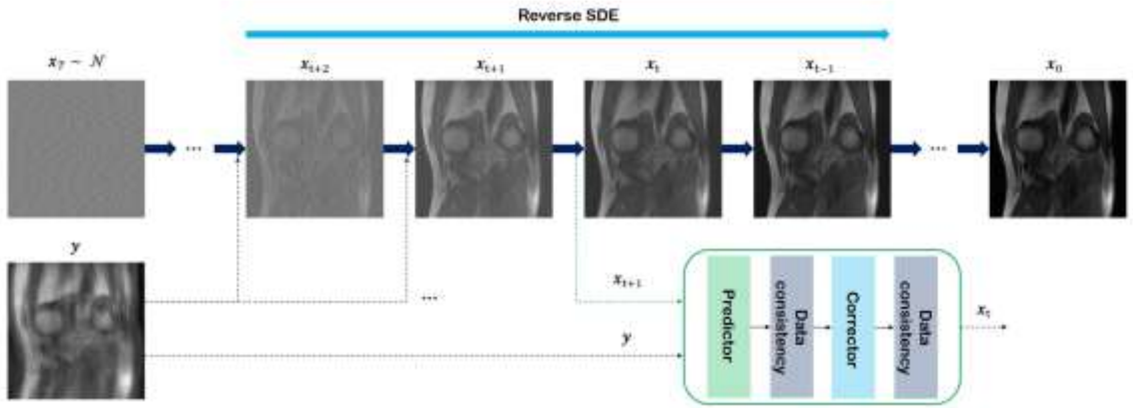
Editing

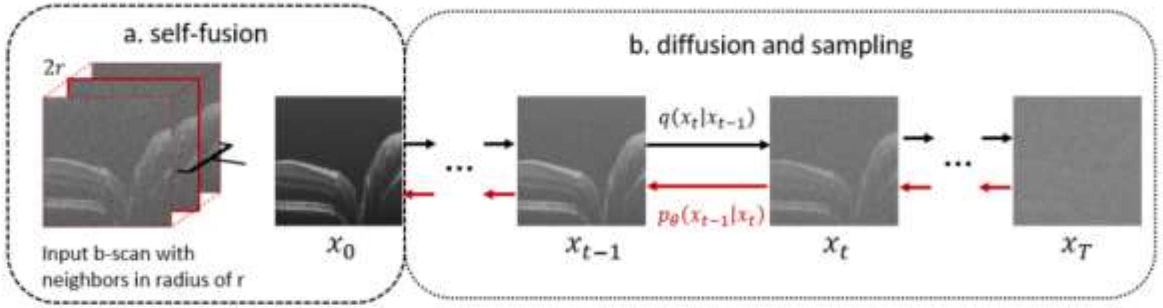
Medical Image Processing: Single Distribution



Classification

Reconstruction

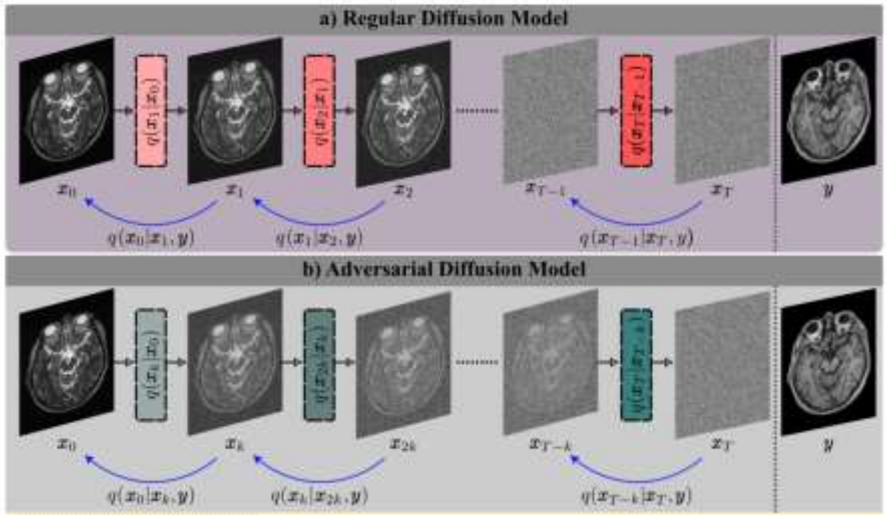


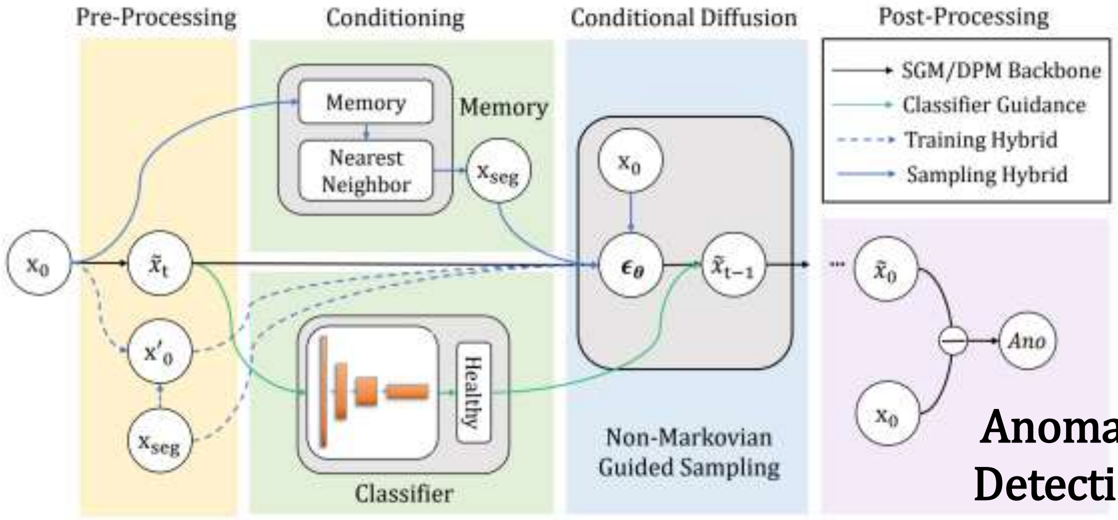


Denoising

Medical Image Processing: I2I Translation

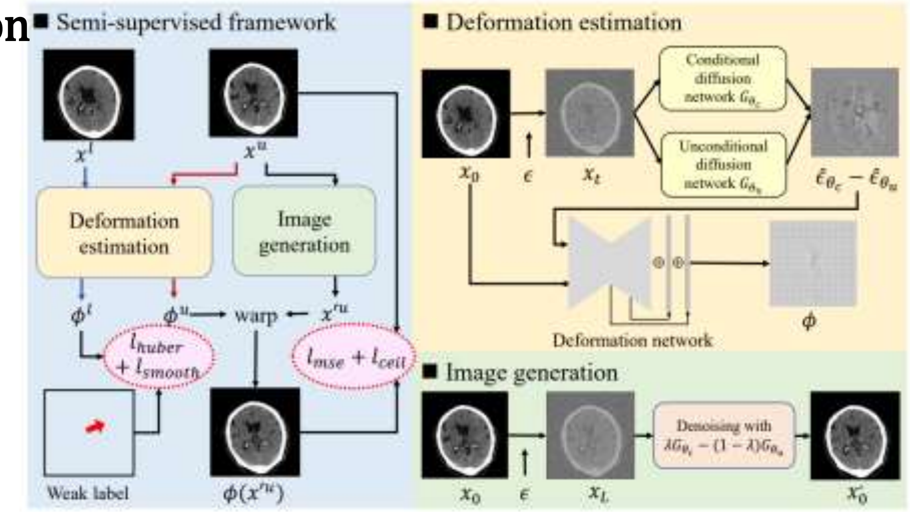
Data Conversion



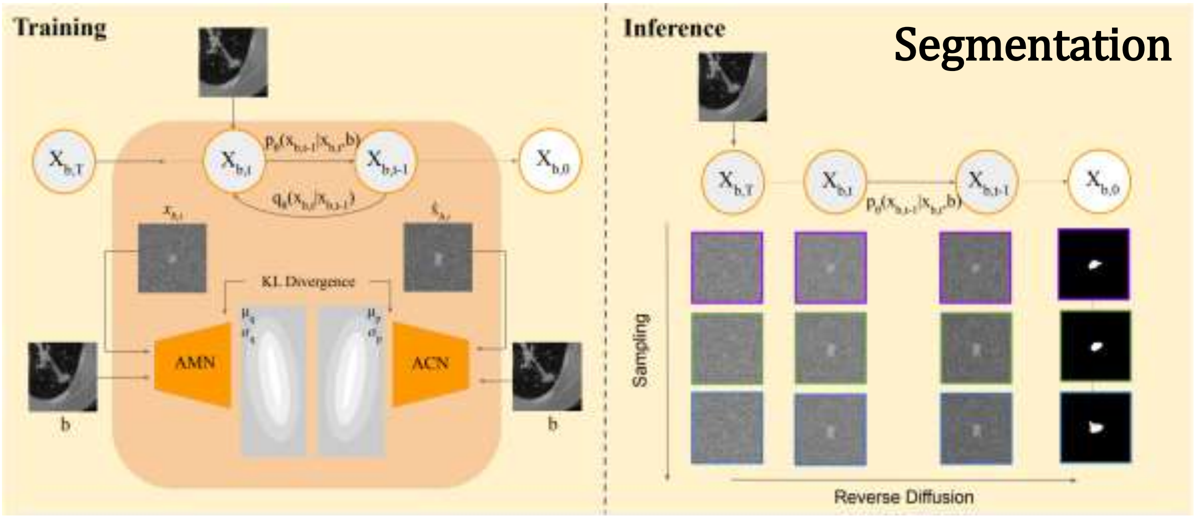


Anomaly Detection

Deformation

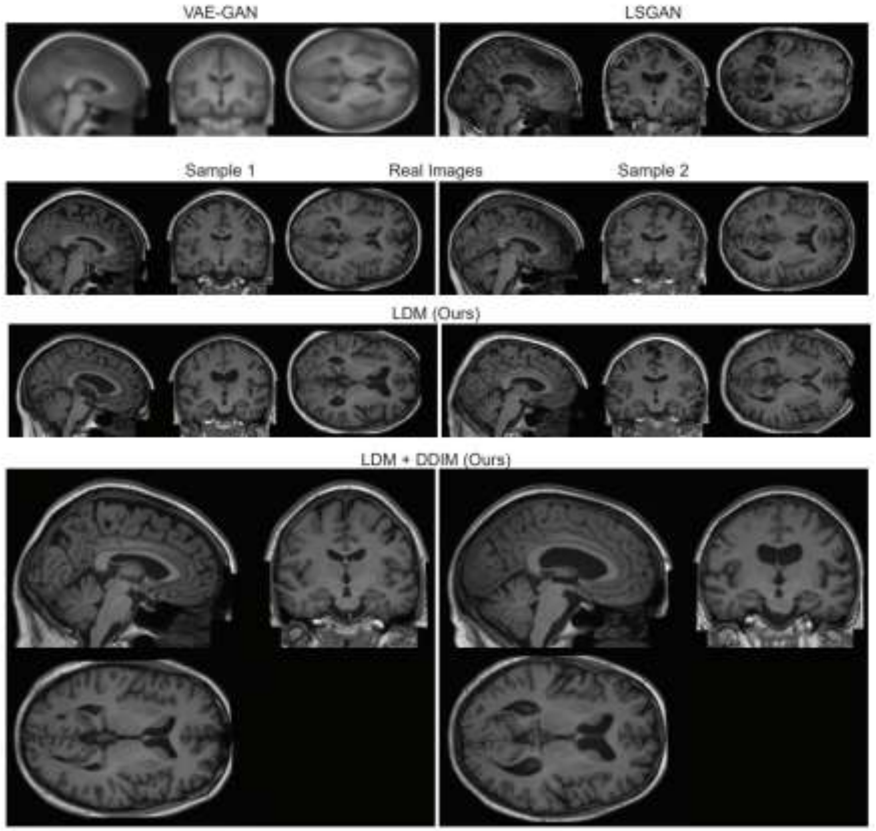


Segmentation

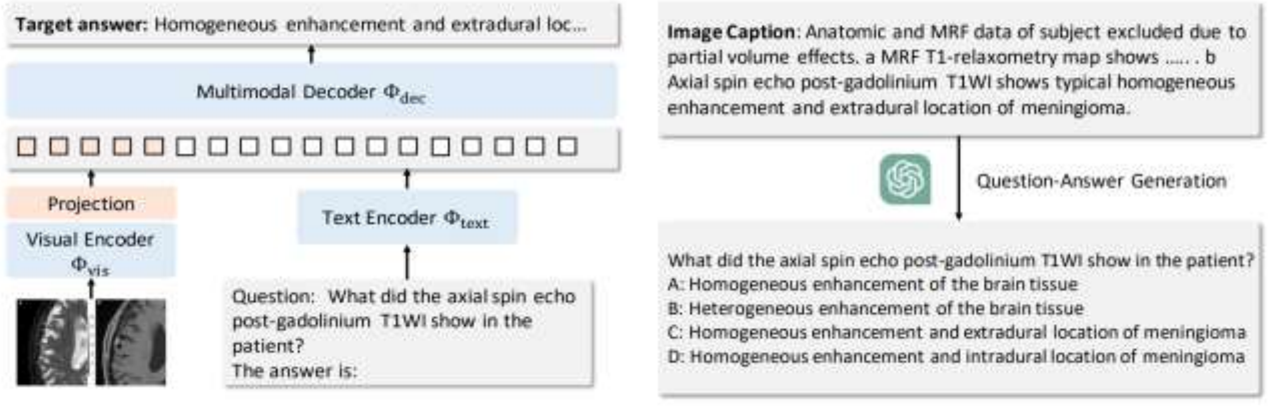


Medical Image Processing: Applications

Multi-Modal System

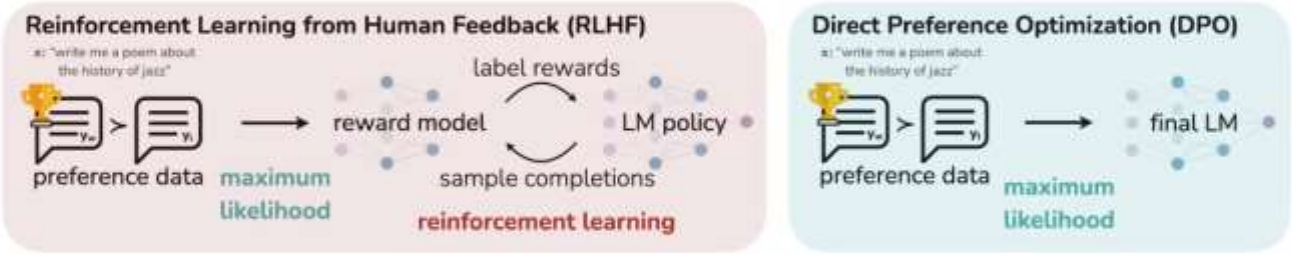


Synthesis



(a) Overall architecture of MedVInT

(b) Pipeline for PMC-VQA generation



Medical GPT

Sequence Modeling: NLP

Discrete Modeling

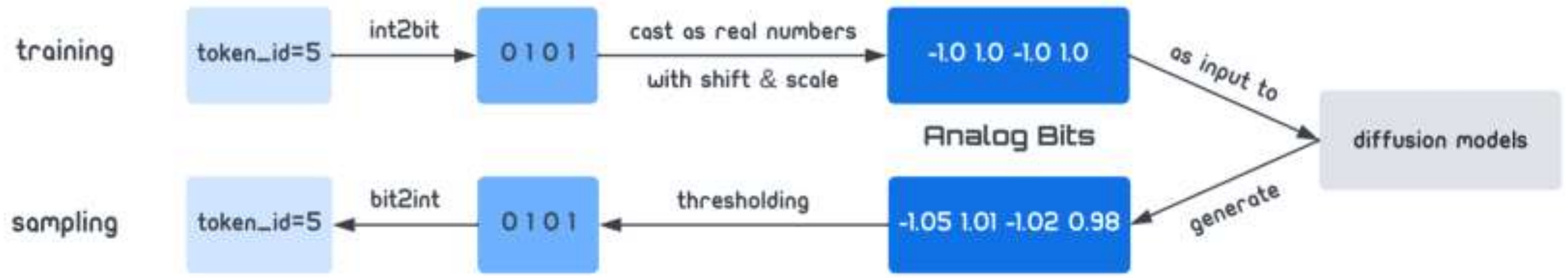
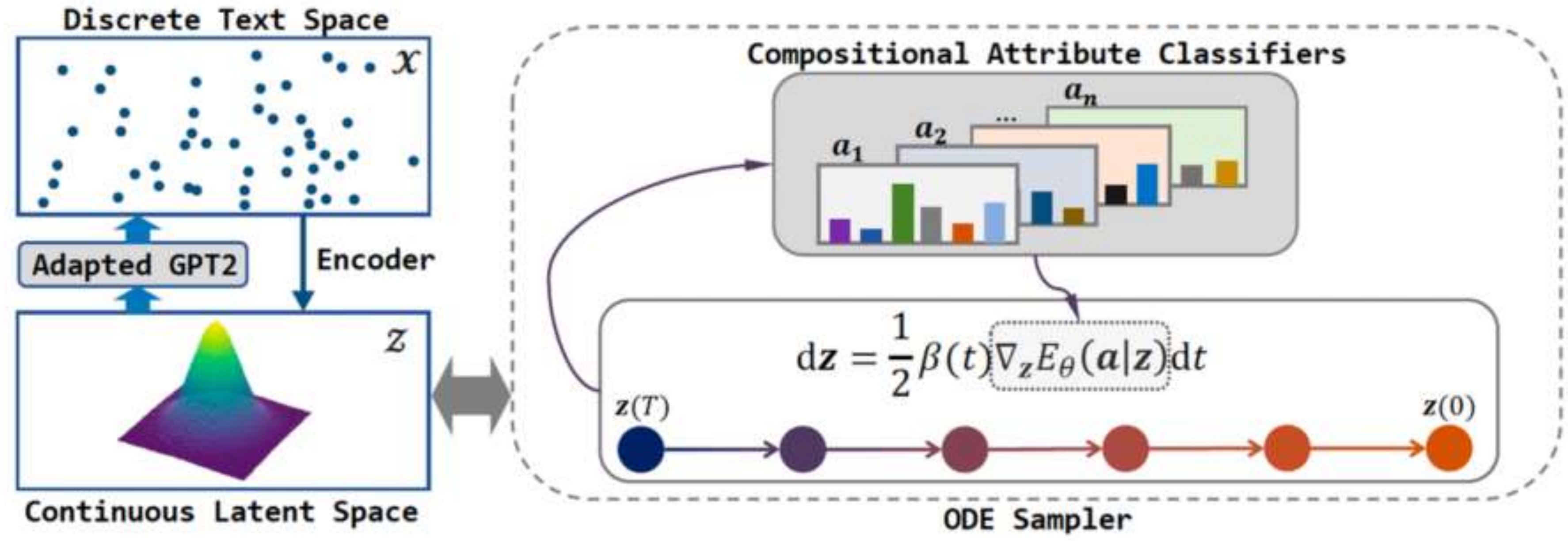


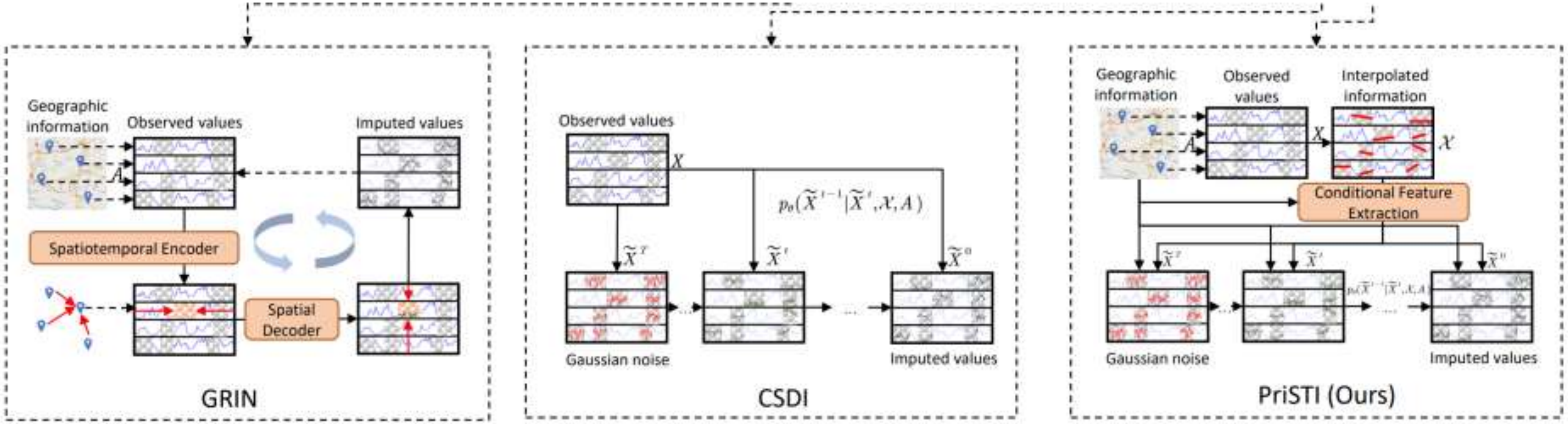
Figure 1: Bit Diffusion: modeling discrete data using continuous diffusion models with analog bits.

Latent Space Generation

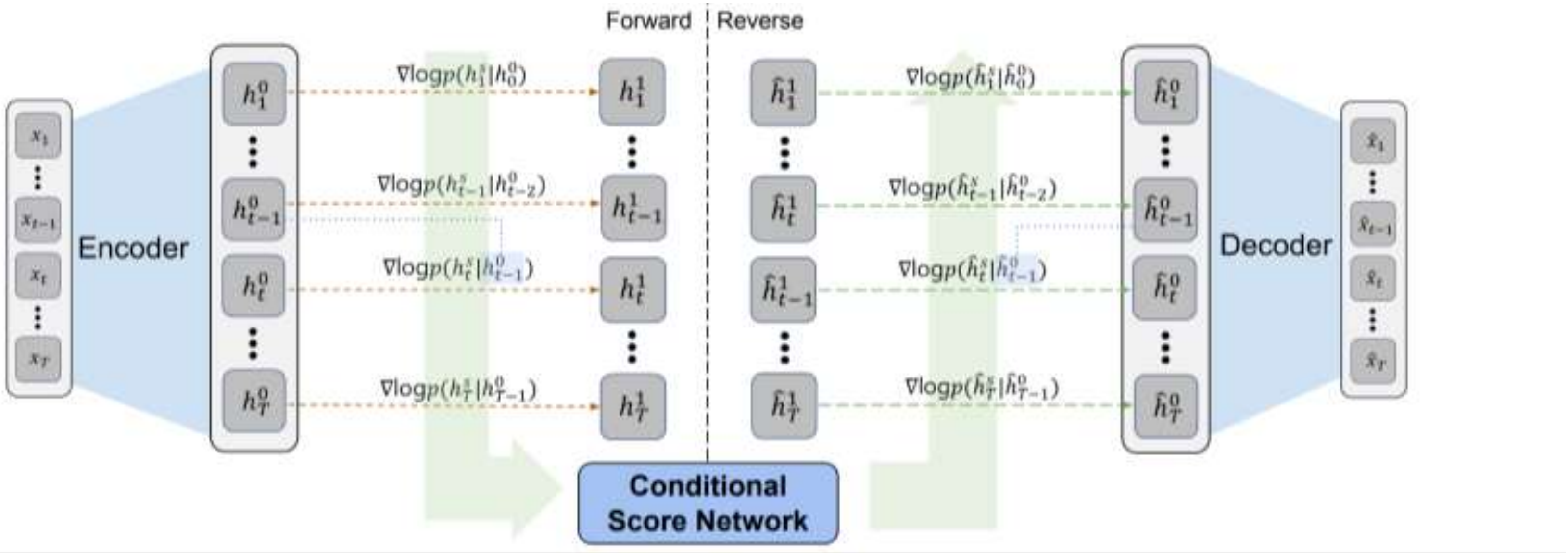


Sequence Modeling: Time Series

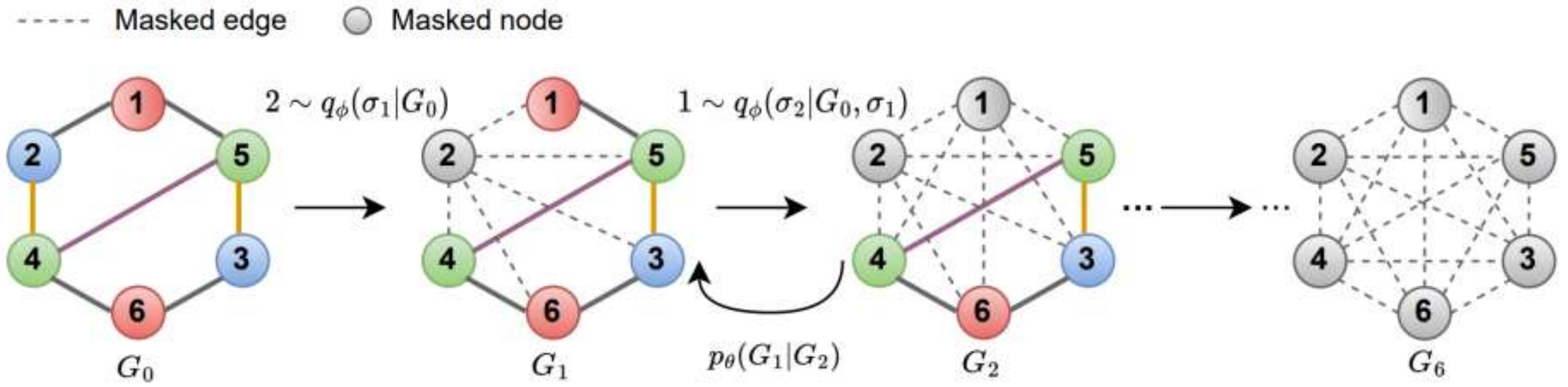
Imputation



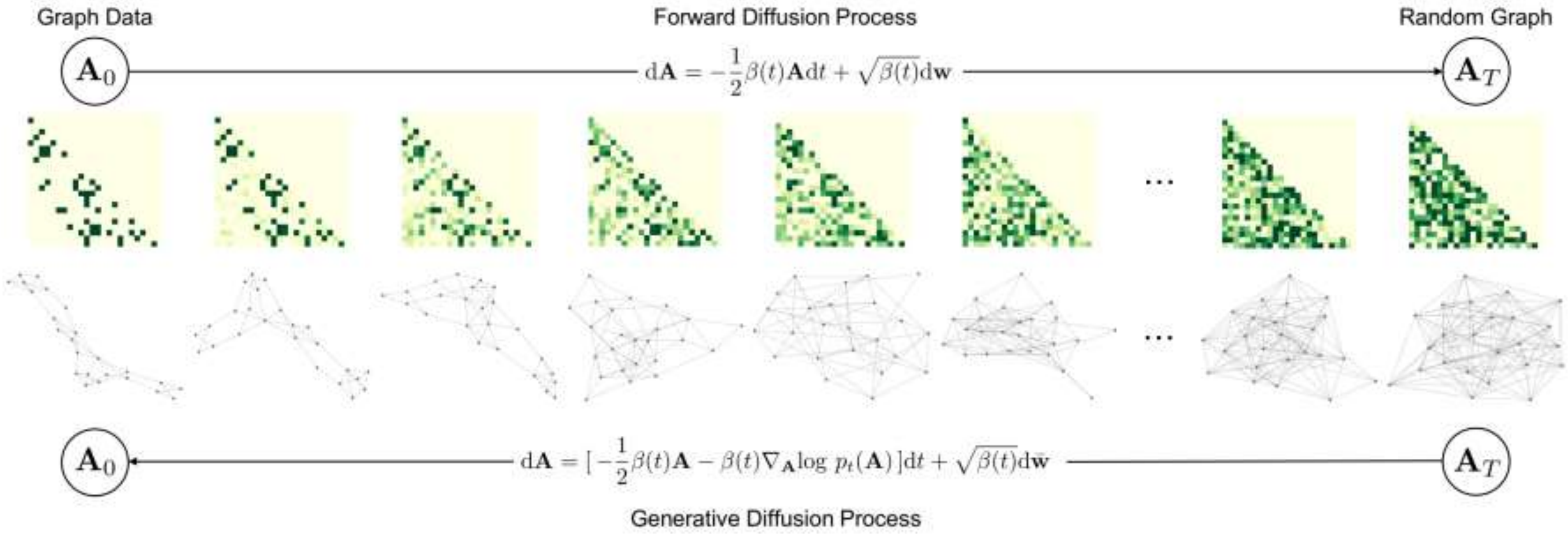
Generation



Graph Modeling: Graph



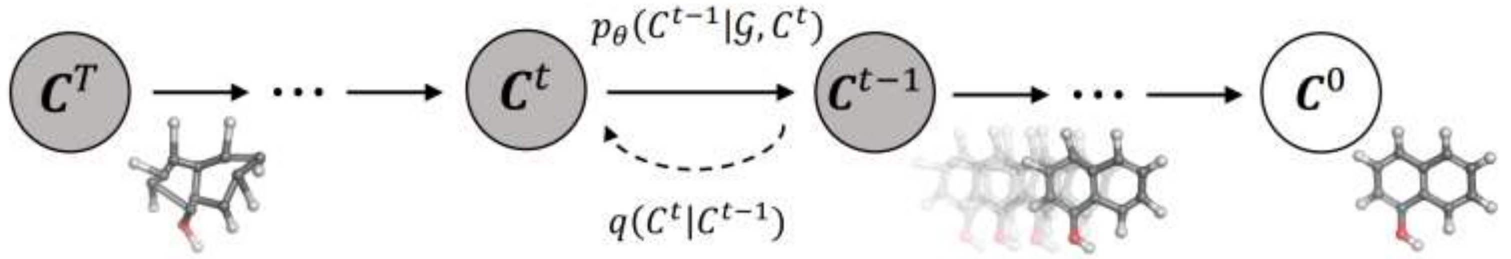
Autogressive Generation



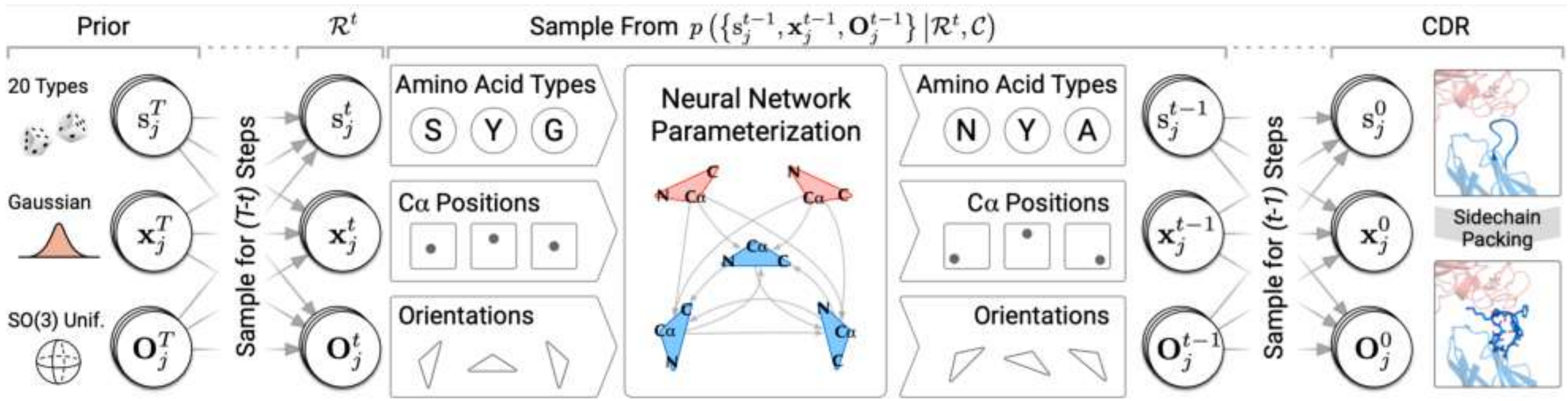
Matrix Generation

Graph Modeling: Molecular Generation

Monomer Design



Complexes Design



Outlines

- Brief Introduction to diffusion model
- Viewing diffusion model through generative model development
- Diffusion model basic algorithms
- Enhancing Understanding from multi-view
- Algorithm improvement
- Applications
- Further Directions and Discussions

How to equip diffusion Model?

Choose your framework

→ ODE: fast, deterministic & controllable generation

→ SDE: slow, high-fidelity and diverse sampling, unconditional generation

Data format & Model Architecture

→Image / Discrete: UNet

→Sequential data: Transformer, LSTM, RNN

→Graph data: Invariant / Equivariant GNNs

How to equip diffusion Model?

Data Amount

→Not enough: Latent diffusion, Data Augmentation, other types of generative model

Sampling Techniques

→Unconditional: depend on the training loss (fast solver or traditional)

→Conditional: classifier-based, pixel-level, latent space

Other Techniques

→ Domain distribution shift, early stop, distillation

Further Directions (Oct 2022):

- Attention on diffusion model class:

  -- Prior distribution, transition kernel, sampling algorithm, and diffusion schemes

- Training objective & evaluation metric:

  -- Evaluation mismatch, Improved objective for MLE

- Application and inductive bias:

  -- Inductive bias, more practice

Further Directions (Oct 2023):

- Generation Quality & Speed:

  -- Advanced distillation on ODE & SDE solvers

- Combined with Large Pre-trained Models:

  -- Act as a re-generator for more diverse samples

- Cross-Modality Generation:

  -- Aligning latent features from multiple modalities

Further Directions (Oct 2023):

- AIGC Era:

  -- Generating highly reliable data for training enhancement

  -- Generating Out-Of-Distribution samples for exploration and attack-defense

- Combined with other fields of ML:

  -- Semi-supervised learning: generated data w./w.o labels ,

  -- Reinforcement learning: reinforcement-guided sampling

  -- Domain Transfer: cross-domain generation

Useful Resources

- Paper source:
https://github.com/diff-usion/Awesome-Diffusion-Models
- Codebases on huggingface:
https://huggingface.co/docs/diffusers/index
- Chinese version of detailed introduction:
https://spaces.ac.cn/author/1/5/
- Great Labs:
https://scholar.google.com/citations?user=axsP38wAAAAJ&hl=zh-CN
https://scholar.google.com/citations?user=Ao4gtsYAAAAJ&hl=en
https://scholar.google.co.uk/citations?user=o_J2CroAAAAJ&hl=en

Finally: Thanks for listening

Paper: https://arxiv.org/abs/2209.02646

GitHub: https://github.com/chq1155/A-Survey-on-Generative-Diffusion-Model

Be a contributor, Involve in diffusion research, conduct diffusion applications!

Thanks for listening
and discussion!