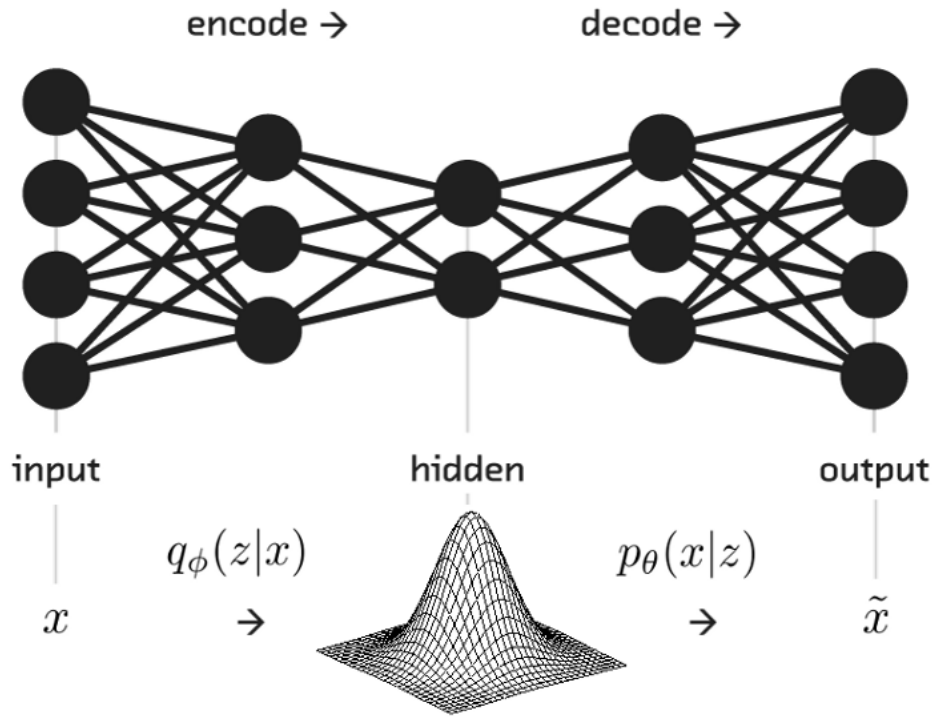
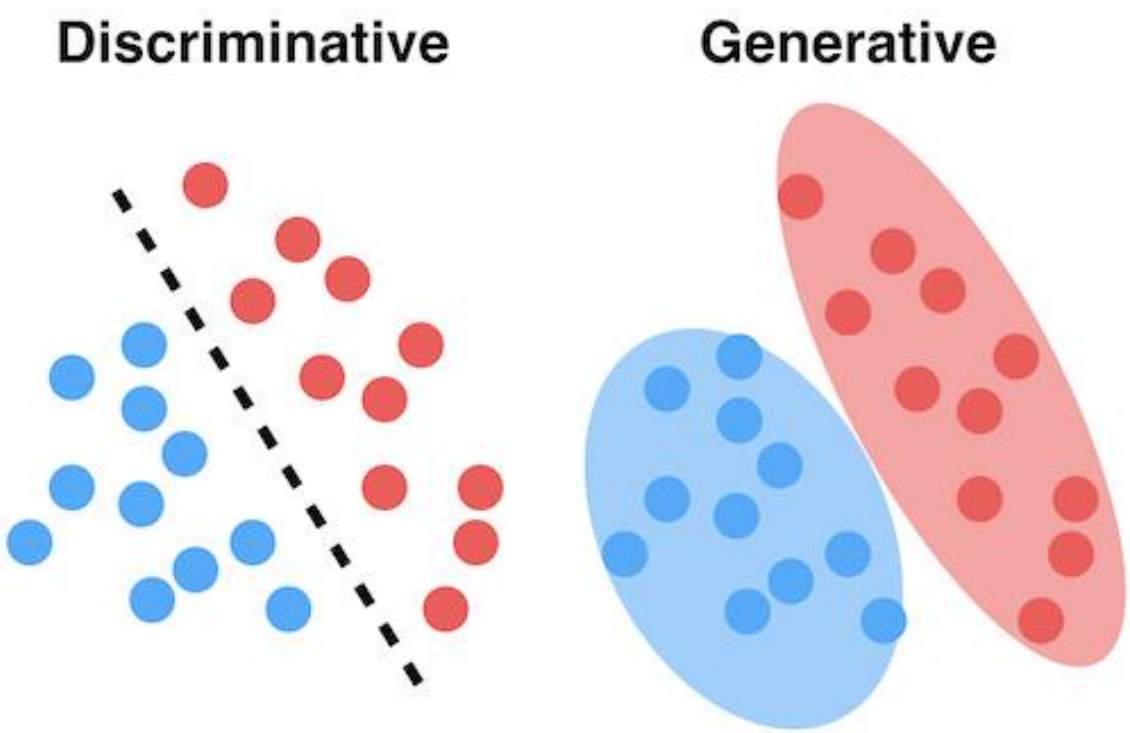
Diffusion Model Review

CAO Hanqun

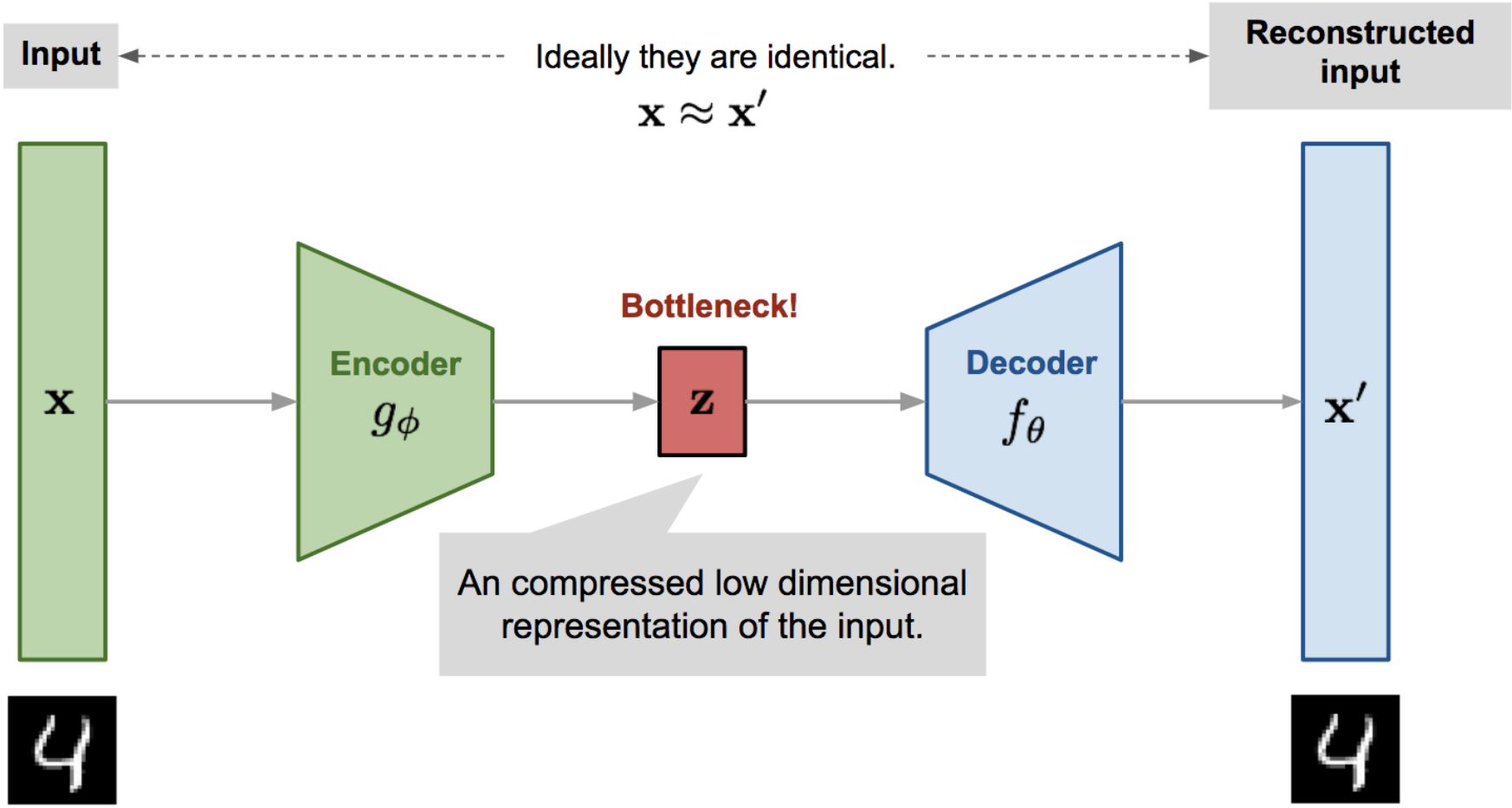
Generative Model

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

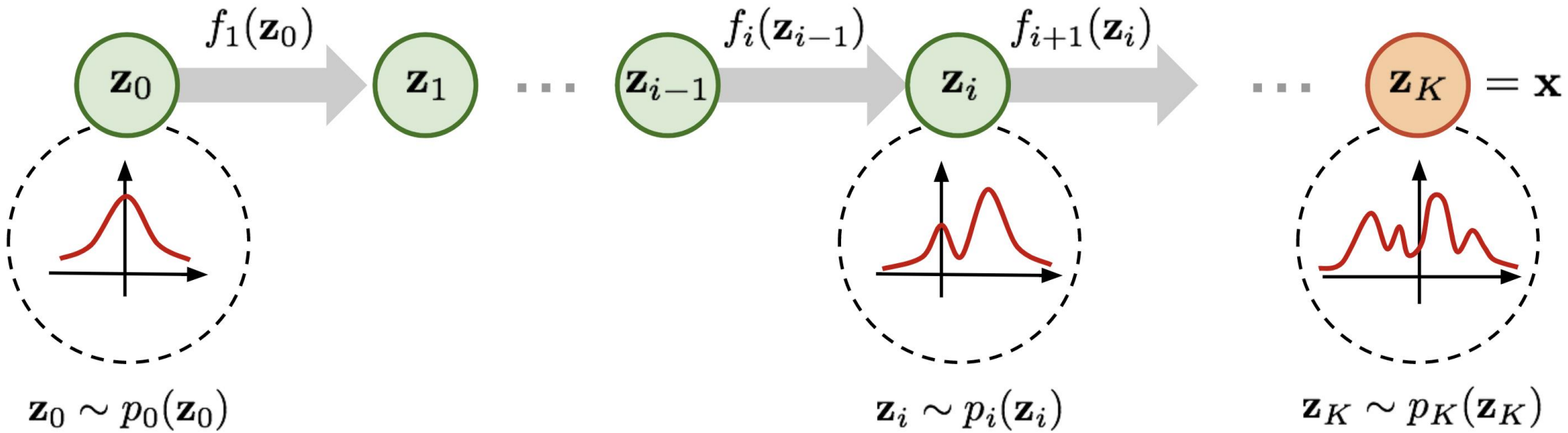




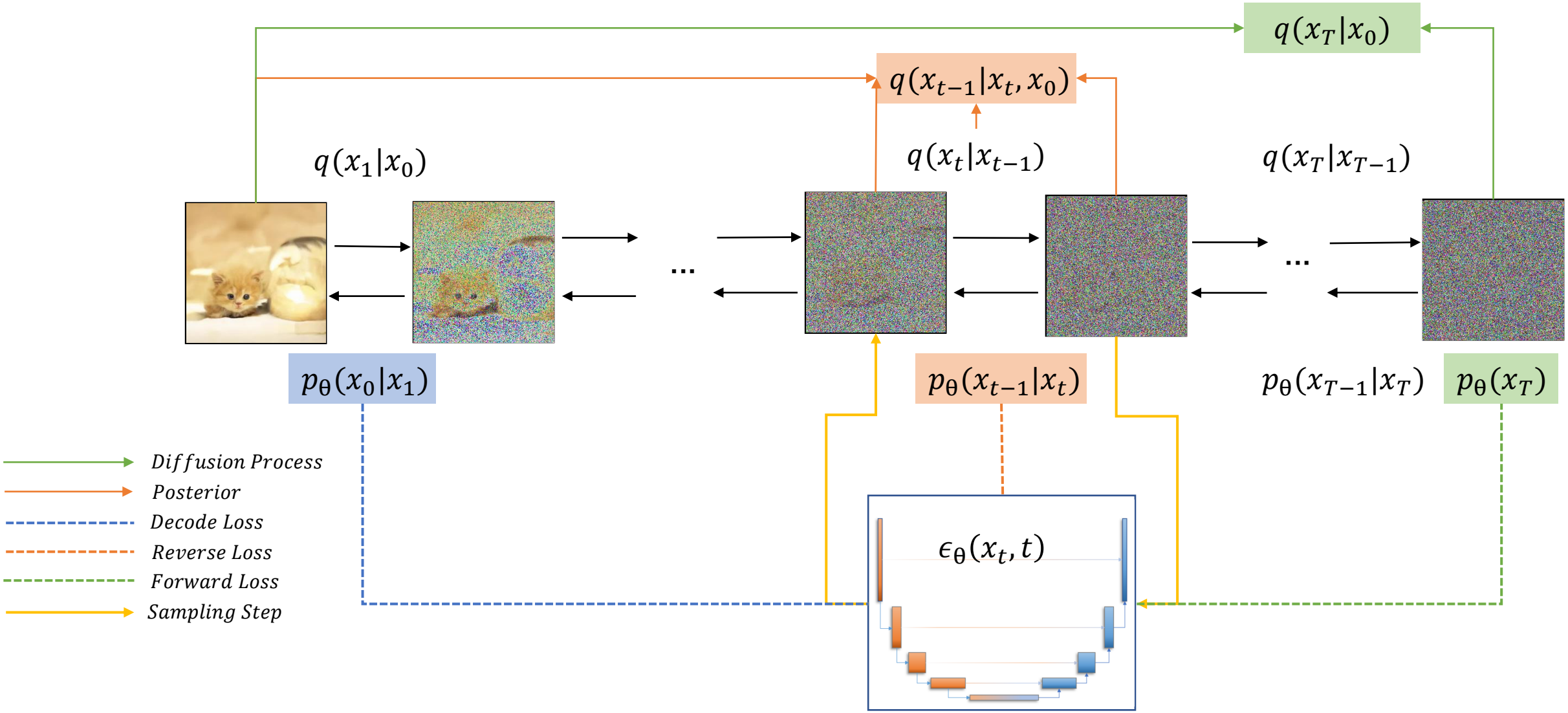
From VAE to NF, to Diffusion Model



From VAE to NF, to Diffusion Model



From VAE to NF, to Diffusion Model



Diffusion Model Math Formulation

- Forward Process

$$F(x,\sigma) = F_T(x_{T-1},\sigma_T)\cdots\circ F_t(x_{t-1},\sigma_t)\cdots\circ F_1(x_0,\sigma_1)$$

- Reverse Process

$$R(x,\sigma) = R_1(x_1,\sigma_1)\cdots\circ R_t(x_t,\sigma_t)\cdots\circ R_T(x_T,\sigma_T)$$

- Transition Kernels

$$x_t = F_t(x_{t-1},\sigma_t), \quad x_{t-1} = R_t(x_t,\sigma_t)$$

$$\tilde{x}_0 = [R_1(x_1,\sigma_1)\cdots\circ R_t(x_t,\sigma_t)\cdots\circ R_T(x_T,\sigma_T)]\circ [F_T(x_{T-1},\sigma_T)\cdots\circ F_t(x_{t-1},\sigma_t)\cdots\circ F_1(x_0,\sigma_1)]$$

- Noising & Denoising Process

$$\mathbb{E}_{F(x_0,\sigma)}\left[-\log R(\mathbf{x}_T,\tilde{\sigma})\right]$$

- Training Objective based on MLE

Landmark Works: From discrete to Continuous

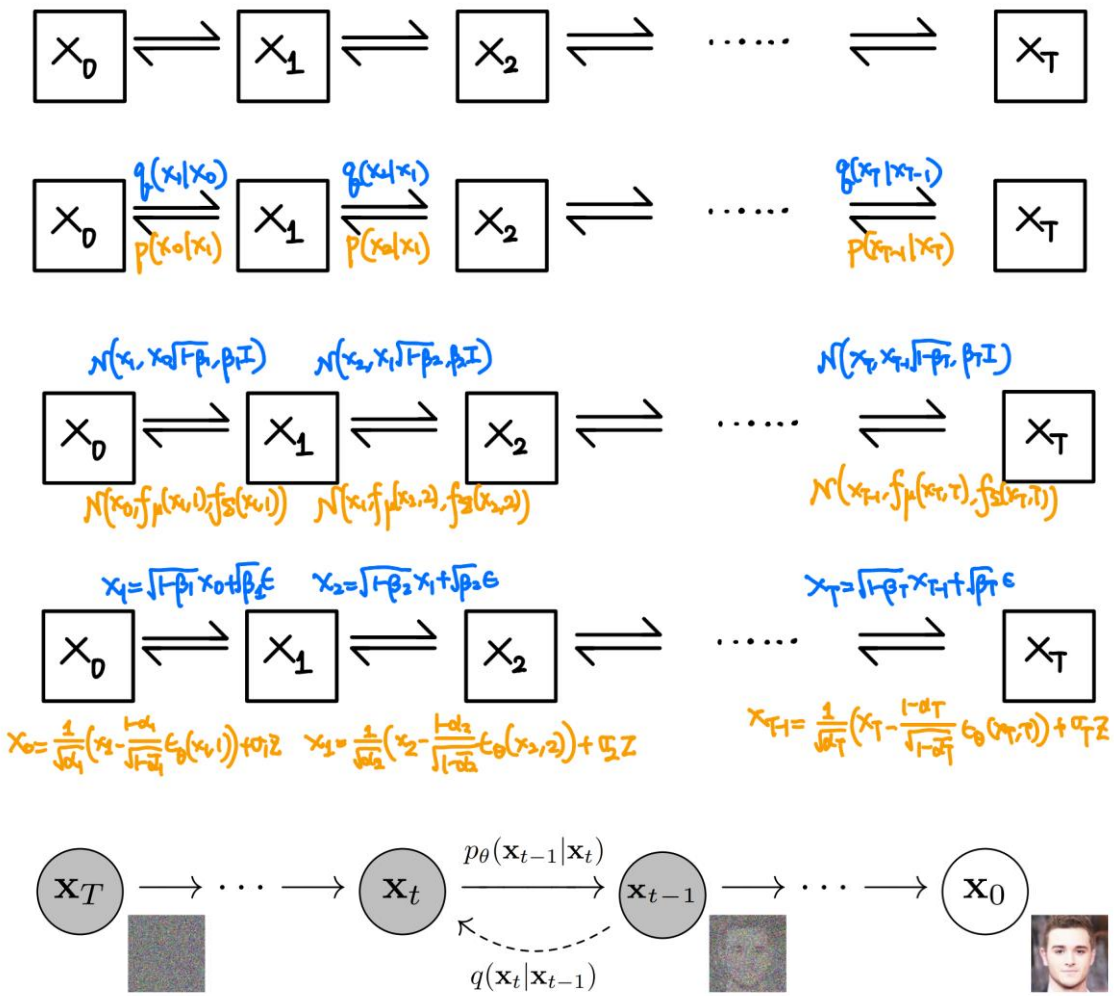


Figure 2: The directed graphical model considered in this work.

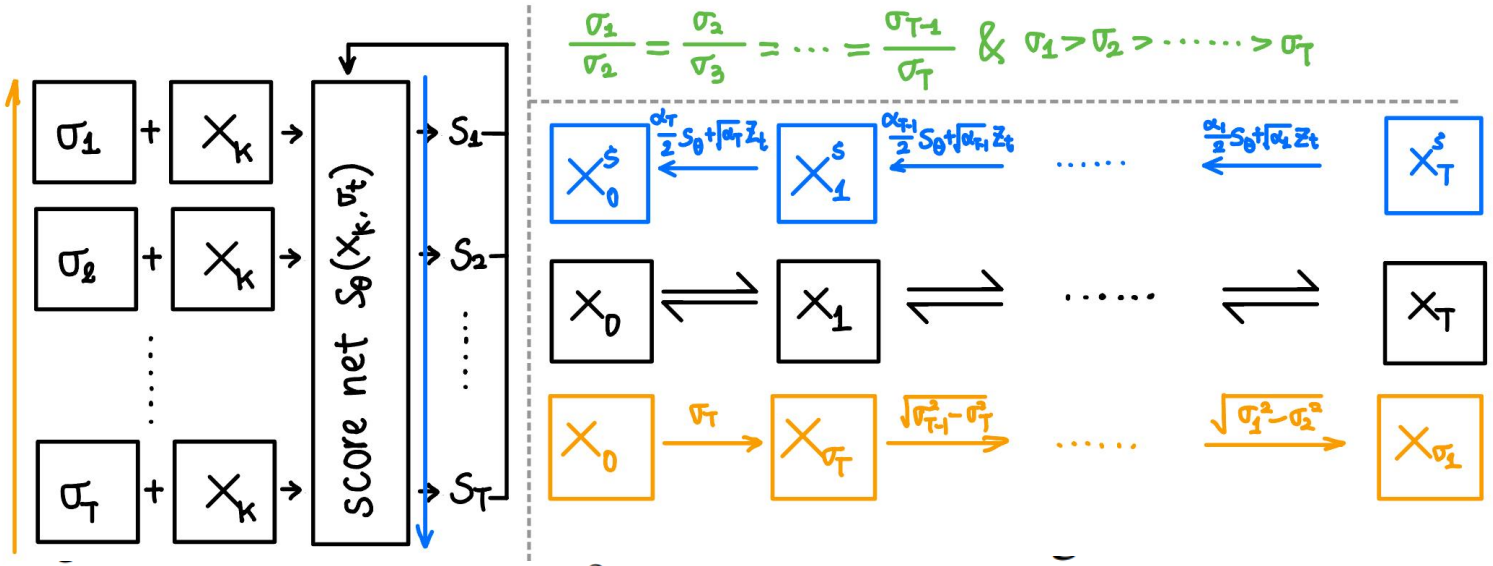
Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
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: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Landmark Works: From discrete to Continuous



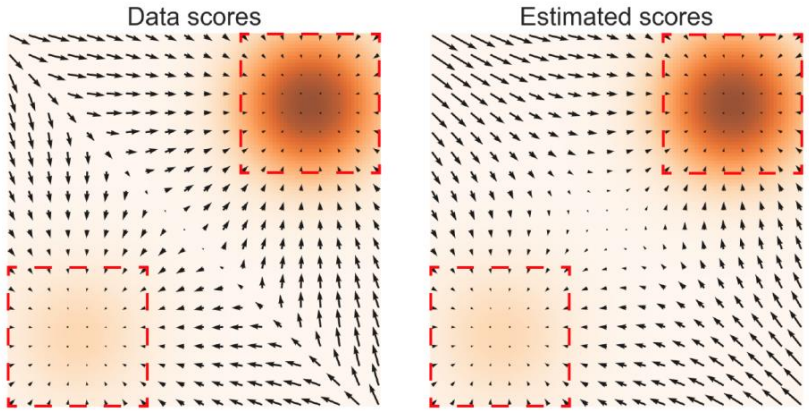
$$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I); \quad \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) = -(\tilde{\mathbf{x}}-\mathbf{x})/\sigma^2$$

$$\ell(\boldsymbol{\theta}; \sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\tilde{\mathbf{x}}\sim\mathcal{N}(\mathbf{x},\sigma^2 I)}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}, \sigma) + \frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2}\right\|_2^2\right].$$

Method	Loss
$\mathcal{L}_{\text{ESM}}$	$\frac{1}{2}\mathbb{E}[\|\|s_\theta(Y_s, s) - \nabla \log q(Y_s)\|\|_\Lambda^2]$
$\mathcal{L}_{\text{ISM}}$	$\mathbb{E}[\frac{1}{2}\|\|s_\theta(Y_s, s)\|\|_\Lambda^2 + \nabla \cdot (\Lambda^\top s_\theta)]$
$\mathcal{L}_{\text{SSM}}$	$\mathbb{E}[\frac{1}{2}\|\|s_\theta(Y_s, s)\|\|_\Lambda^2 + v^\top \nabla(\Lambda^\top s_\theta)v]$
$\mathcal{L}_{\text{DSM}}$	$\frac{1}{2}\mathbb{E}[\|\|s_\theta(Y_s, s) - \nabla \log q(Y_s\|Y_0)\|\|_\Lambda^2]$

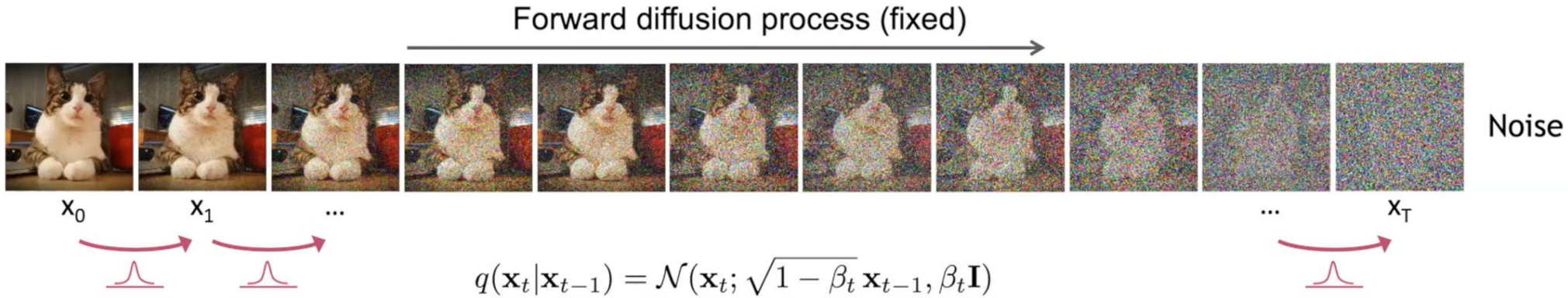
Algorithm 1 Annealed Langevin dynamics [1]

Require: $\{\sigma_i\}_{i=1}^L, \epsilon, T$.
1: Initialize $\mathbf{x}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad \alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad$ Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
6: $\quad\quad$ $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \alpha_i\, \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}, \sigma_i) + \sqrt{2\alpha_i}\, \mathbf{z}_t$
7: $\quad$ $\mathbf{x}_0 \leftarrow \mathbf{x}_T$
8: **if** denoise $\mathbf{x}_T$ **then**
9: $\quad$ **return** $\mathbf{x}_T + \sigma_T^2 \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_T, \sigma_T)$
10: **else**
11: $\quad$ **return** $\mathbf{x}_T$



Landmark Works: From discrete to Continuous

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

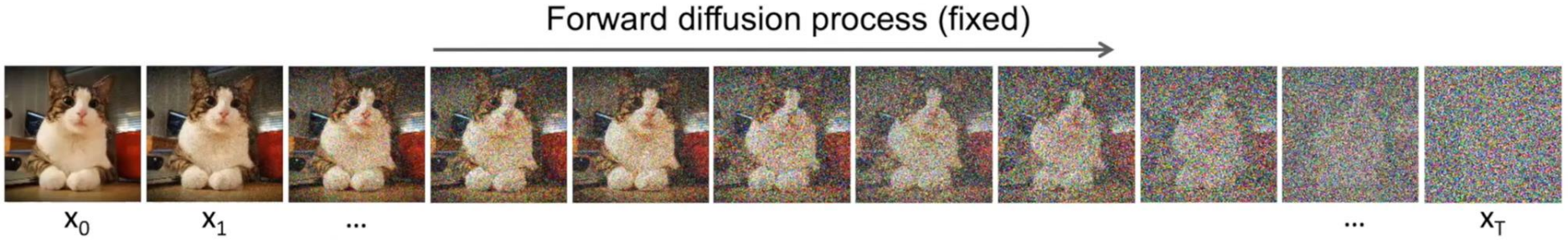


$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

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

Landmark Works: From discrete to Continuous

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

Landmark Works: From discrete to Continuous

Forward SDE (data → noise)

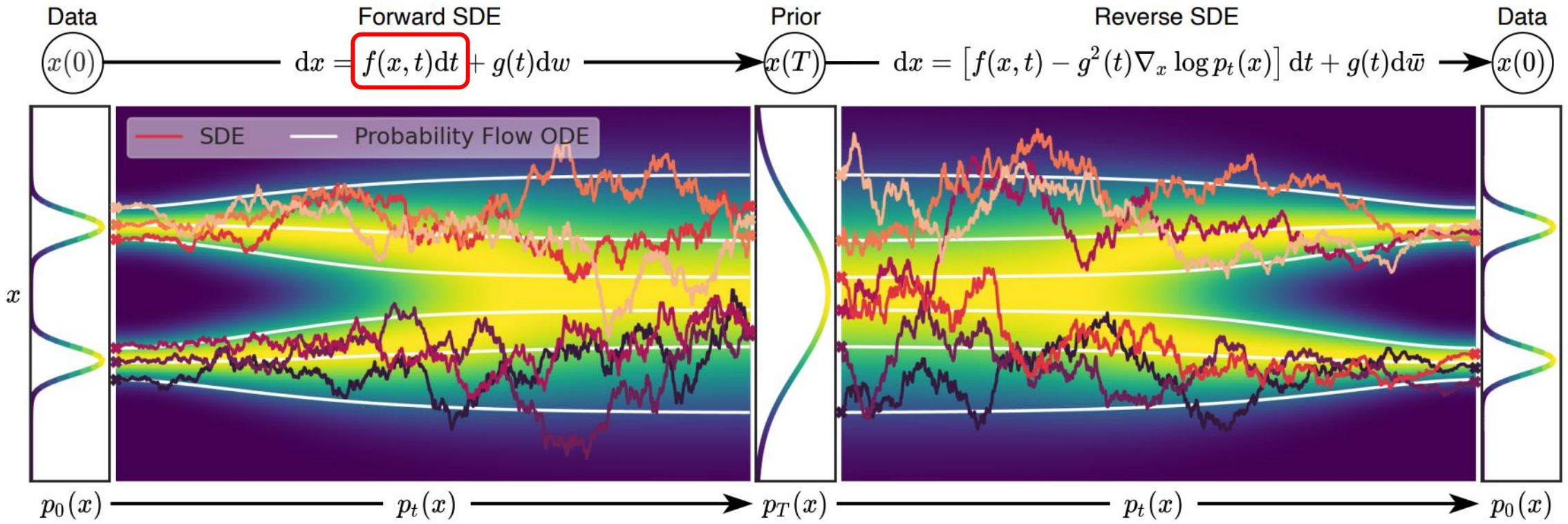
$\mathbf{x}(0)$ ——— $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ ———→ $\mathbf{x}(T)$

score function

$\mathbf{x}(0)$ ←——— $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ ——— $\mathbf{x}(T)$

Reverse SDE (noise → data)

Landmark Works: From discrete to Continuous



- SDE: Higher Performance
- ODE: Higher Speed

Landmark Works: From discrete to Continuous

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

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

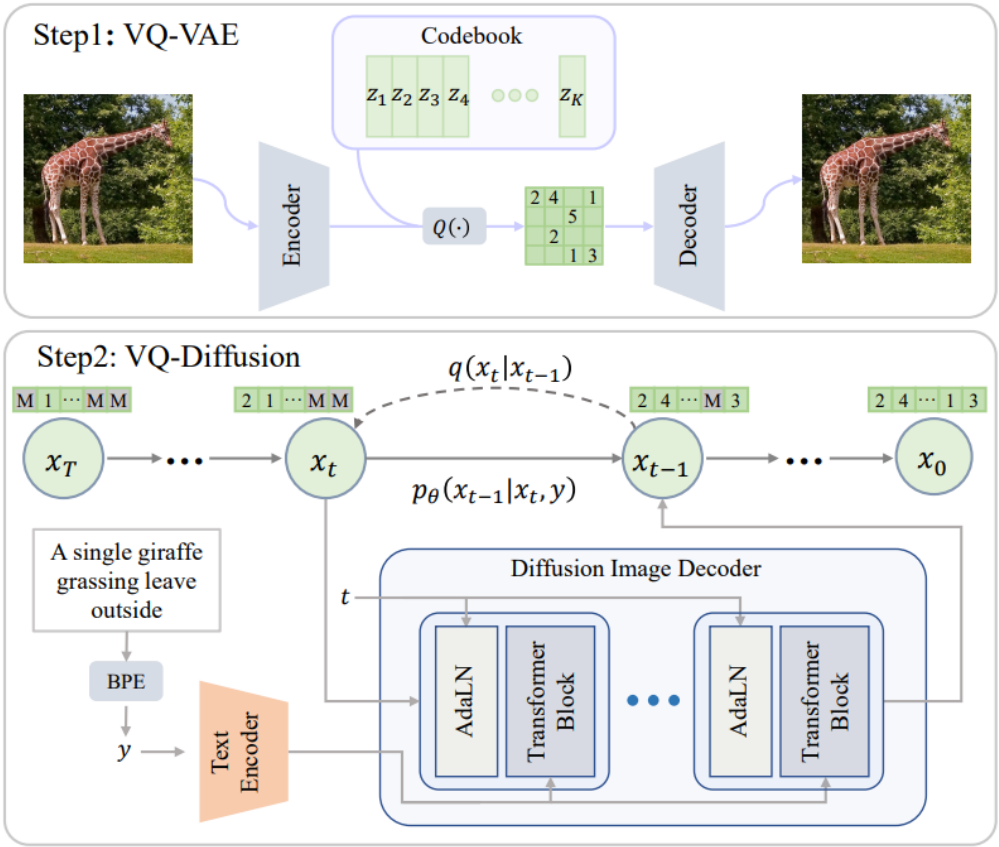
-- Sampling along the reverse trajectory is actually finding the numerical solution of differential equations.

Landmark Works: Label_Conditional Diffusion

Diffusion Models Beat GANs on Image Synthesis

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.

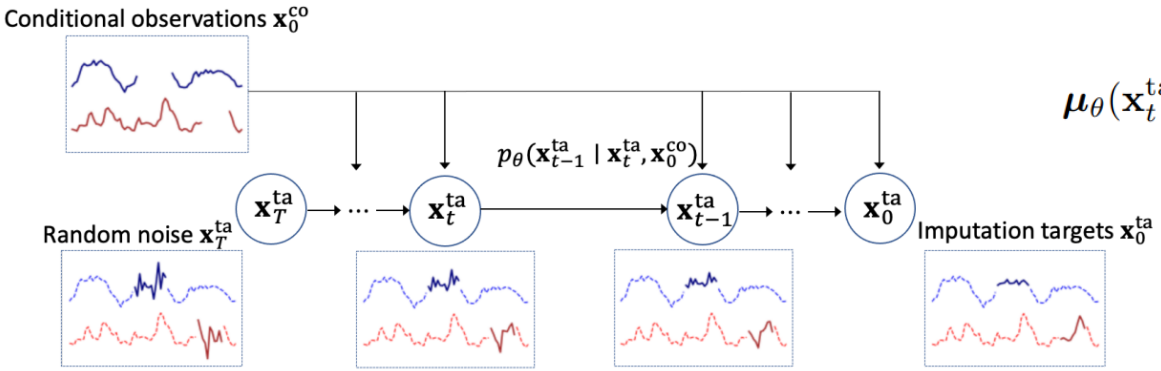
Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for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



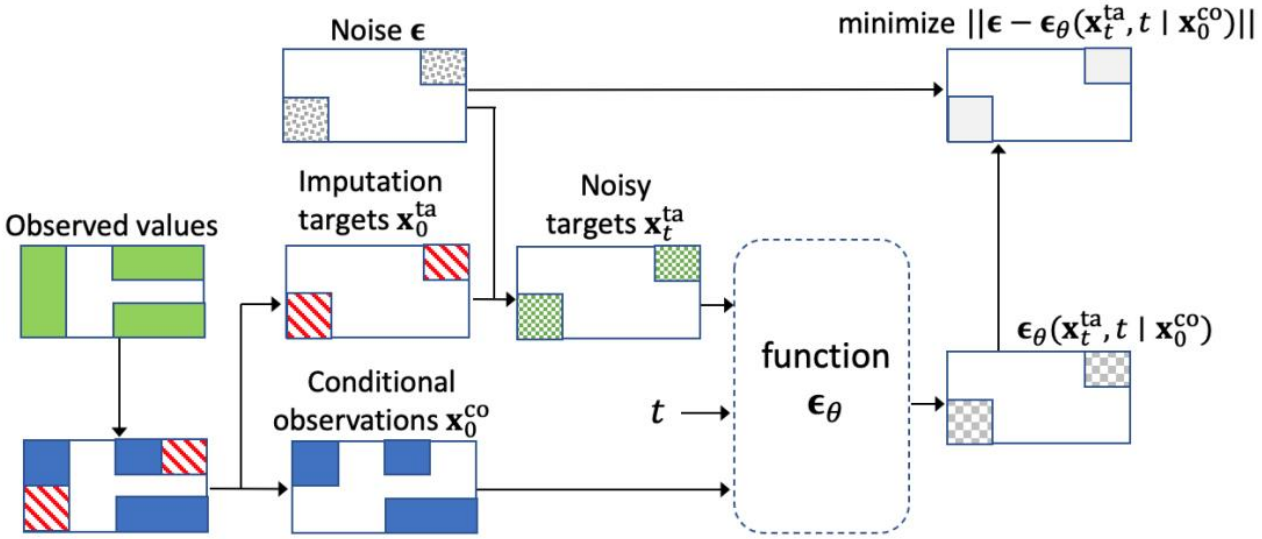
Score-Based Generative Modeling through Stochastic Differential Equations

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

Landmark Works: Data_Conditional Diffusion



$$\boldsymbol{\mu}_\theta(\mathbf{x}_t^{\mathrm{ta}}, t \mid \mathbf{x}_0^{\mathrm{co}}) = \boldsymbol{\mu}^{\mathrm{DDPM}}(\mathbf{x}_t^{\mathrm{ta}}, t, \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{ta}}, t \mid \mathbf{x}_0^{\mathrm{co}})), \quad \sigma_\theta(\mathbf{x}_t^{\mathrm{ta}}, t \mid \mathbf{x}_0^{\mathrm{co}}) = \sigma^{\mathrm{DDPM}}(\mathbf{x}_t^{\mathrm{ta}}, t)$$

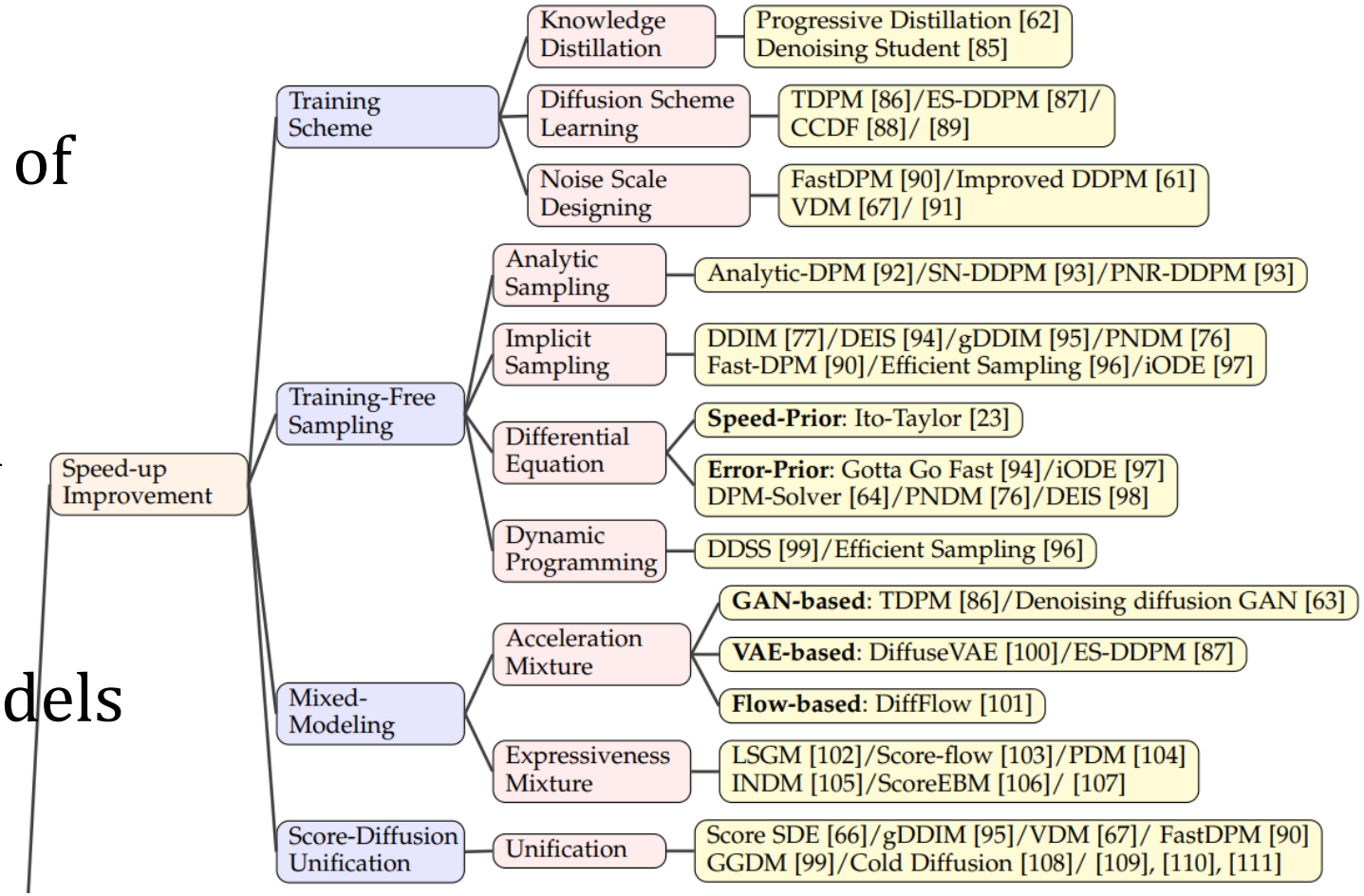


$$\mathcal{L}(\theta) := \min_\theta \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} ||(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{ta}}, t \mid \mathbf{x}_0^{\mathrm{co}}))||_2^2$$

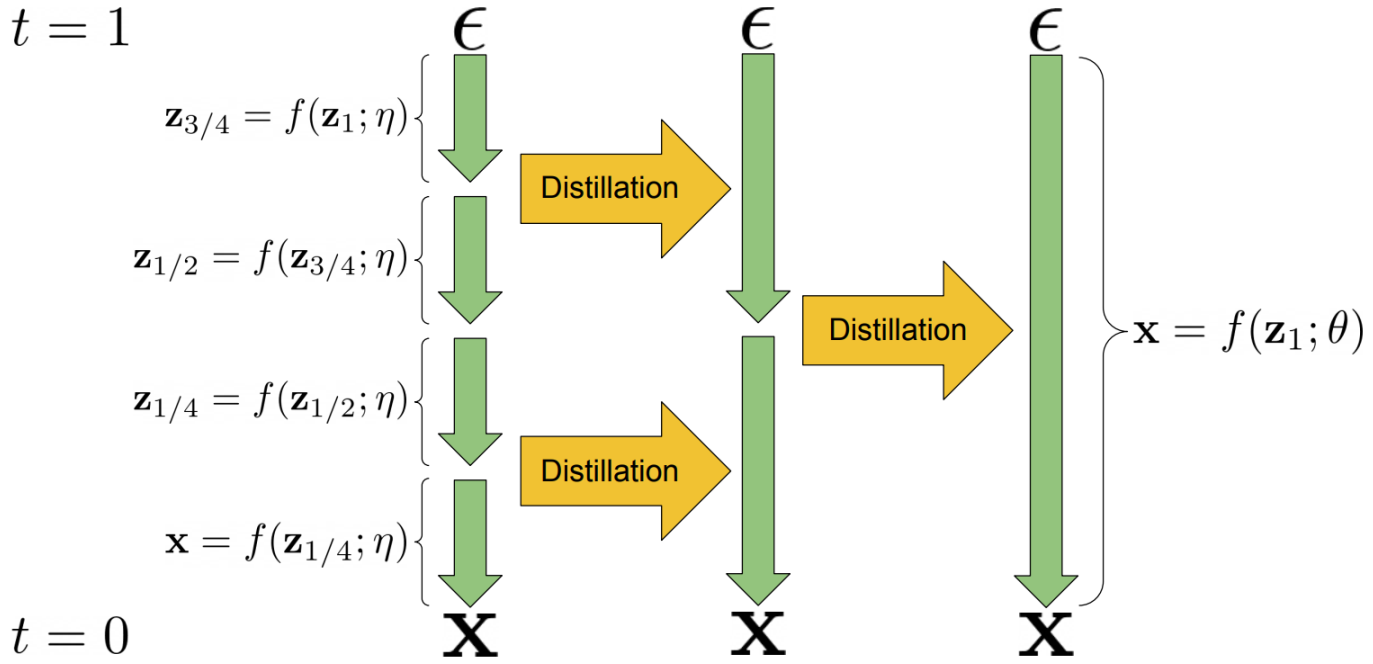
Diffusion Needs Improvements

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

1. Change diffusion pattern
2. Advanced sampling
3. Seek help from other models
4. Unified Framework



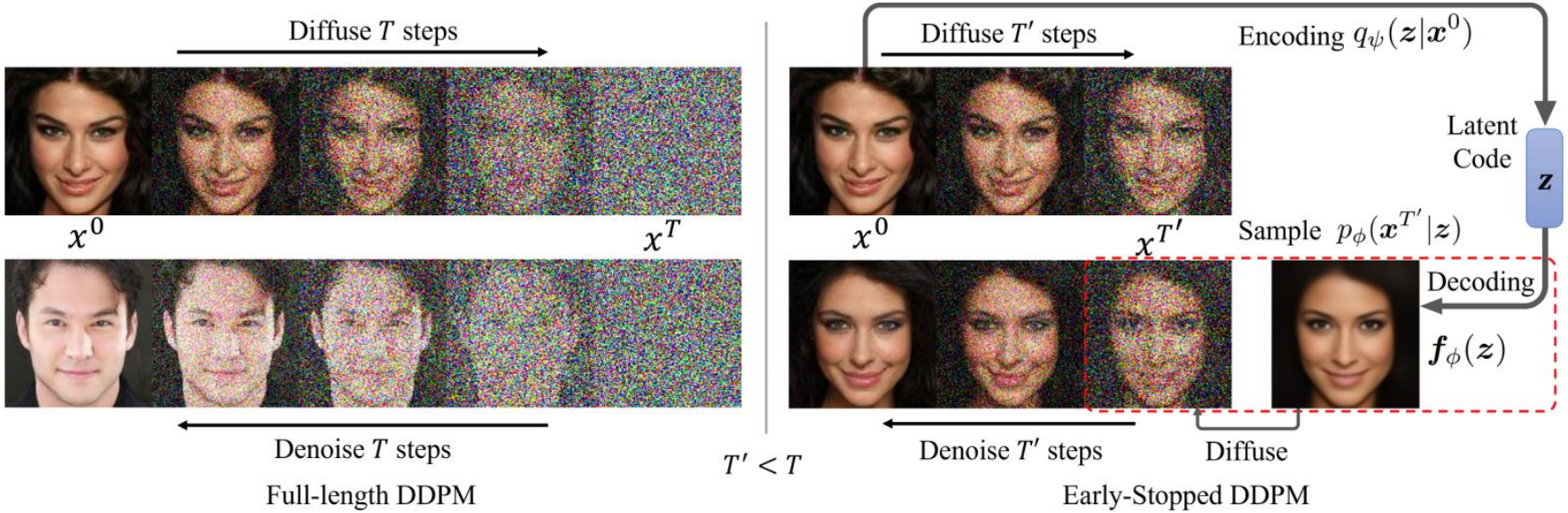
Training Scheme: Distillation



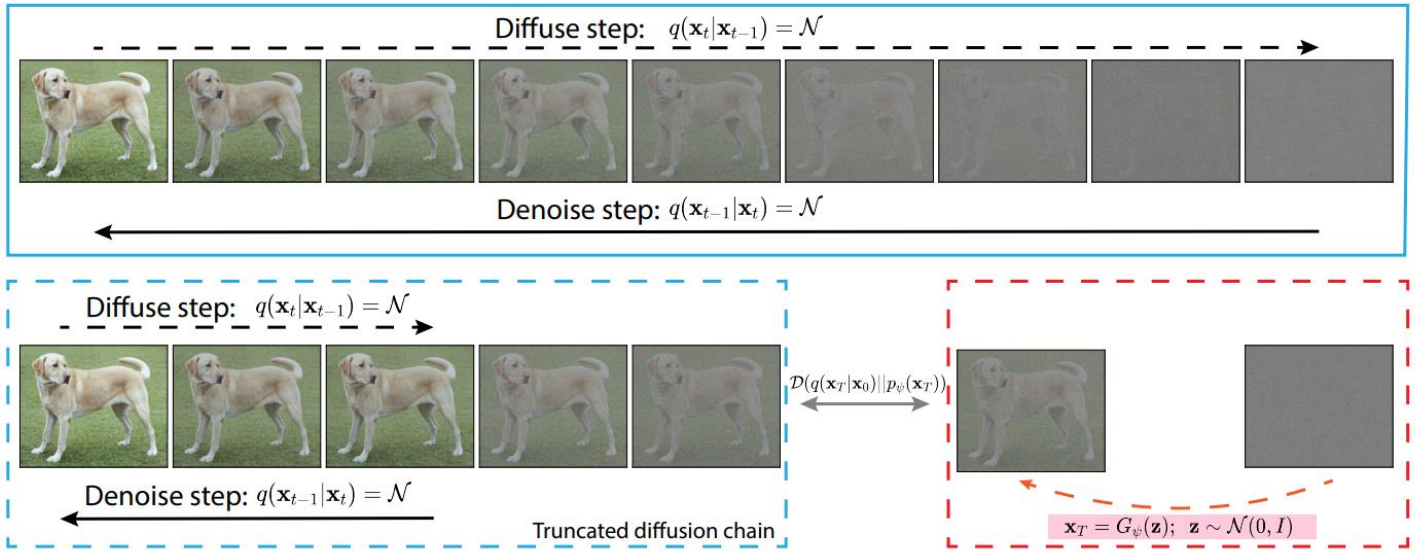
Algorithm 2 Progressive distillation

Require: Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$
Require: Student sampling steps N
  for K iterations **do**
    $\theta \leftarrow \eta$ ▷ Init student from teacher
    while not converged **do**
      $\mathbf{x} \sim \mathcal{D}$
      $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
      $\epsilon \sim N(0, I)$
      $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
      `# 2 steps of DDIM with teacher`
      $t' = t - 0.5/N, \;\; t'' = t - 1/N$
      $\mathbf{z}_{t'} = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t))$
      $\mathbf{z}_{t''} = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}_{t'} - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}))$
      $\tilde{\mathbf{x}} = \frac{\mathbf{z}_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
      $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$
      $L_\theta = w(\lambda_t)\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$
      $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$
    end while
    $\eta \leftarrow \theta$ ▷ Student becomes next teacher
    $N \leftarrow N/2$ ▷ Halve number of sampling steps
  end for

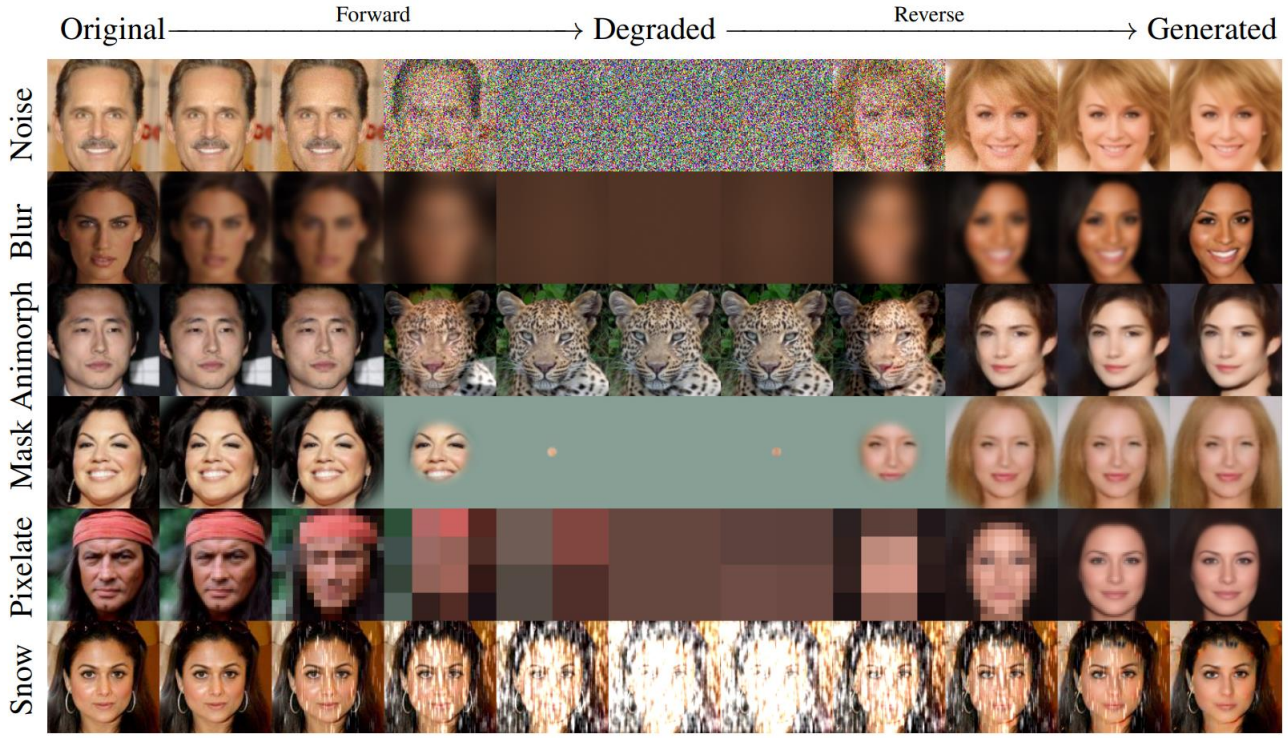
Training Scheme: Stop!



Early Stop!



Training Scheme: Exploring the distribution



Algorithm 1 Naive Sampling

Input: A degraded sample x_t
for $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = D(\hat{x}_0, s-1)$
end for
Return: x_0

Algorithm 2 Improved Sampling for Cold Diffusion

Input: A degraded sample x_t
for $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = x_s - D(\hat{x}_0, s) + D(\hat{x}_0, s-1)$
end for

Training Scheme: Design the Noise

- 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

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

guide the training with SNR:

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

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

Algorithm 2 Sampling

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

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

Advanced Sampling: Analytic Solver

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*

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

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

- 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 3.2. *(Optimal solution to optimization solely w.r.t. mean) For any covariance* $\boldsymbol{\Sigma}_n$*, the optimal mean to problem (12) is always Eq. (4), i.e.,*

$$\boldsymbol{\mu}_n^*(\boldsymbol{x}_n) = \tilde{\boldsymbol{\mu}}_n\left(\boldsymbol{x}_n, \frac{1}{\sqrt{\overline{\alpha}_n}}(\boldsymbol{x}_n - \sqrt{\overline{\beta}_n}\mathbb{E}_{q(\boldsymbol{x}_0|\boldsymbol{x}_n)}[\boldsymbol{\epsilon}_n])\right),$$

which is irrelevant to $\boldsymbol{\Sigma}_n$*.*

Theorem 3.3. *(Optimal solution to optimization w.r.t. covariance solely) Suppose* $\boldsymbol{\Sigma}_n(\boldsymbol{x}_n) = \mathrm{diag}(\boldsymbol{\sigma}_n(\boldsymbol{x}_n)^2)$*. For any mean* $\boldsymbol{\mu}_n(\boldsymbol{x}_n)$ *that is parameterized by a noise prediction network* $\hat{\boldsymbol{\epsilon}}_n(\boldsymbol{x}_n)$ *as in Eq. (6), the optimal covariance* $\tilde{\boldsymbol{\sigma}}_n^*(\boldsymbol{x}_n)^2$ *to problem (13) is*

$$\begin{aligned}\tilde{\boldsymbol{\sigma}}_n^*(\boldsymbol{x}_n)^2 &= \boldsymbol{\sigma}_n^*(\boldsymbol{x}_n)^2 + \gamma_n^2 \frac{\overline{\beta}_n}{\overline{\alpha}_n} \underbrace{(\hat{\boldsymbol{\epsilon}}_n(\boldsymbol{x}_n) - \mathbb{E}_{q(\boldsymbol{x}_0|\boldsymbol{x}_n)}[\boldsymbol{\epsilon}_n])^2}_{error} \\ &= \lambda_n^2 \mathbf{1} + \gamma_n^2 \frac{\overline{\beta}_n}{\overline{\alpha}_n} \mathbb{E}_{q(\boldsymbol{x}_0|\boldsymbol{x}_n)}[(\boldsymbol{\epsilon}_n - \hat{\boldsymbol{\epsilon}}_n(\boldsymbol{x}_n))^2], \quad (14)\end{aligned}$$

where $\boldsymbol{\sigma}_n^*(\boldsymbol{x}_n)^2$ *is the optimal covariance to the joint optimization problem in Theorem 3.1,* $\boldsymbol{\epsilon}_n = \frac{\boldsymbol{x}_n - \sqrt{\overline{\alpha}_n}\boldsymbol{x}_0}{\sqrt{\overline{\beta}_n}}$ *is the noise used to generate* $\boldsymbol{x}_n$ *from* $\boldsymbol{x}_0$*,* $(\cdot)^2$ *is the element-wise square,* $\mathbf{1}$ *is the vector of ones and* $\gamma_n = \sqrt{\overline{\alpha}_{n-1}} - \sqrt{\overline{\beta}_{n-1} - \lambda_n^2}\sqrt{\frac{\overline{\alpha}_n}{\overline{\beta}_n}}$*.*

Advanced Sampling: Implicit/non-Gaussian

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

$$\text{where} \quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \quad \text{and} \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

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

Comparison between DDPM and DDIM

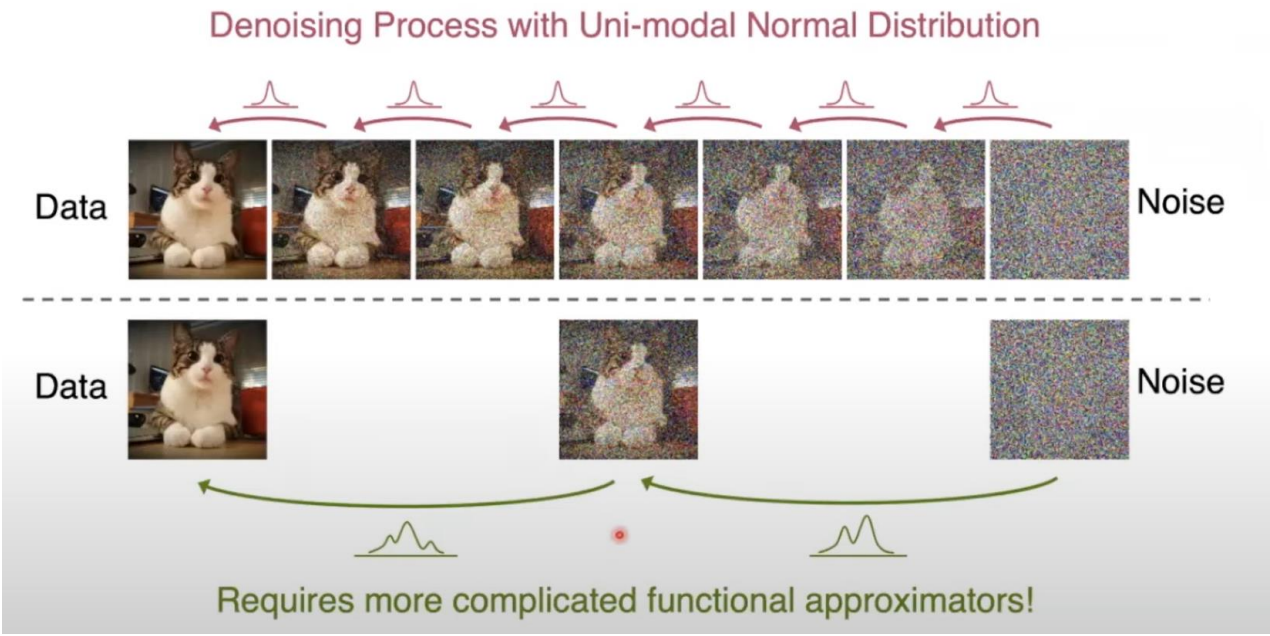
- Implicit Sampling

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

- Implicit Sampling as continuous neural ODE

$$\frac{\boldsymbol{x}_{t-\Delta t}}{\sqrt{\alpha_{t-\Delta t}}} = \frac{\boldsymbol{x}_t}{\sqrt{\alpha_t}} + \left(\sqrt{\frac{1-\alpha_{t-\Delta t}}{\alpha_{t-\Delta t}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}}\right)\epsilon_\theta^{(t)}(\boldsymbol{x}_t)$$

$$\mathrm{d}\bar{\boldsymbol{x}}(t) = \epsilon_\theta^{(t)}\left(\frac{\bar{\boldsymbol{x}}(t)}{\sqrt{\sigma^2+1}}\right)\mathrm{d}\sigma(t),$$



Advanced Sampling: Differential Equation Solver

- Difference between ODE-based and SDE-based models
- Corresponding solver for ODE and SDE
- Difference between higher order and lower order solvers

Improved Techniques:

1. Mid-point solver
2. Multi-methods solver
3. Threshold
4. Dynamic step size adjustment

Algorithm 2 Our stochastic sampler with $\sigma(t) = t$ and $s(t) = 1$.

procedure StochasticSampler($D_\theta(\boldsymbol{x};\sigma)$, $t_{i\in\{0,\dots,N\}}$, $\gamma_{i\in\{0,\dots,N-1\}}$, S_{noise})
  sample $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0},\ t_0^2\,\mathbf{I})$
  for $i \in \{0,\dots,N-1\}$ **do** ▷ $\gamma_i = \begin{cases} \min\left(\frac{S_{\text{churn}}}{N}, \sqrt{2}-1\right) & \text{if } t_i \in [S_{\text{tmin}}, S_{\text{tmax}}] \\ 0 & \text{otherwise} \end{cases}$
    sample $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0},\ S_{\text{noise}}^2\,\mathbf{I})$
    $\hat{t}_i \leftarrow t_i + \gamma_i t_i$ ▷ Select temporarily increased noise level $\hat{t}_i$
    $\hat{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_i + \sqrt{\hat{t}_i^2 - t_i^2}\,\boldsymbol{\epsilon}_i$ ▷ Add new noise to move from t_i to $\hat{t}_i$
    $\boldsymbol{d}_i \leftarrow \left(\hat{\boldsymbol{x}}_i - D_\theta(\hat{\boldsymbol{x}}_i;\hat{t}_i)\right)/\hat{t}_i$ ▷ Evaluate $\mathrm{d}\boldsymbol{x}/\mathrm{d}t$ at $\hat{t}_i$
    $\boldsymbol{x}_{i+1} \leftarrow \hat{\boldsymbol{x}}_i + (t_{i+1} - \hat{t}_i)\boldsymbol{d}_i$ ▷ Take Euler step from $\hat{t}_i$ to t_{i+1}
    if $t_{i+1} \neq 0$ **then**
      $\boldsymbol{d}'_i \leftarrow \left(\boldsymbol{x}_{i+1} - D_\theta(\boldsymbol{x}_{i+1};t_{i+1})\right)/t_{i+1}$ ▷ Apply 2nd order correction
      $\boldsymbol{x}_{i+1} \leftarrow \hat{\boldsymbol{x}}_i + (t_{i+1} - \hat{t}_i)\left(\frac{1}{2}\boldsymbol{d}_i + \frac{1}{2}\boldsymbol{d}'_i\right)$
  return $\boldsymbol{x}_N$

Algorithm 1 Dynamic step size extrapolation for solving Reverse Diffusion Processes

Require: $s_\theta, \epsilon_{rel}, \epsilon_{abs}, h_{init} = 0.01, r = 0.9, \theta = 0.9$ ▷ for images: $\epsilon_{abs} = \frac{y_{max} - y_{min}}{256}$
  $t \leftarrow 1$
  $h \leftarrow h_{init}$
  Initialize $\mathbf{x}$
  $\mathbf{x}'_{prev} \leftarrow \mathbf{x}$
  while $t > 0$ **do**
    Draw $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}' \leftarrow \mathbf{x} - hf(\mathbf{x},t) + hg(t)^2 s_\theta(\mathbf{x},t) + \sqrt{h}g(t)\mathbf{z}$ ▷ Euler-Maruyama
    $\tilde{\mathbf{x}} \leftarrow \mathbf{x} - hf(\mathbf{x}', t-h) + hg(t-h)^2 s_\theta(\mathbf{x}', t-h) + \sqrt{h}g(t-h)\mathbf{z}$
    $\mathbf{x}'' \leftarrow \frac{1}{2}(\mathbf{x}' + \tilde{\mathbf{x}})$ ▷ Improved Euler (SDE version)
    $\boldsymbol{\delta} \leftarrow \max(\epsilon_{abs}, \epsilon_{rel} \max(|\mathbf{x}'|, |\mathbf{x}'_{prev}|))$ ▷ Element-wise operations
    $E_2 \leftarrow \frac{1}{\sqrt{n}} \left\| (\mathbf{x}' - \mathbf{x}'') / \boldsymbol{\delta} \right\|_2$
    if $E_2 \leq 1$ **then** ▷ Accept
      $\mathbf{x} \leftarrow \mathbf{x}''$ ▷ Extrapolation
      $t \leftarrow t - h$
      $\mathbf{x}'_{prev} \leftarrow \mathbf{x}'$
    $h \leftarrow \min(t,\ \theta h E_2^{-r})$ ▷ Dynamic step size update
  return $\mathbf{x}$

Advanced Sampling: Dynamic Sampling

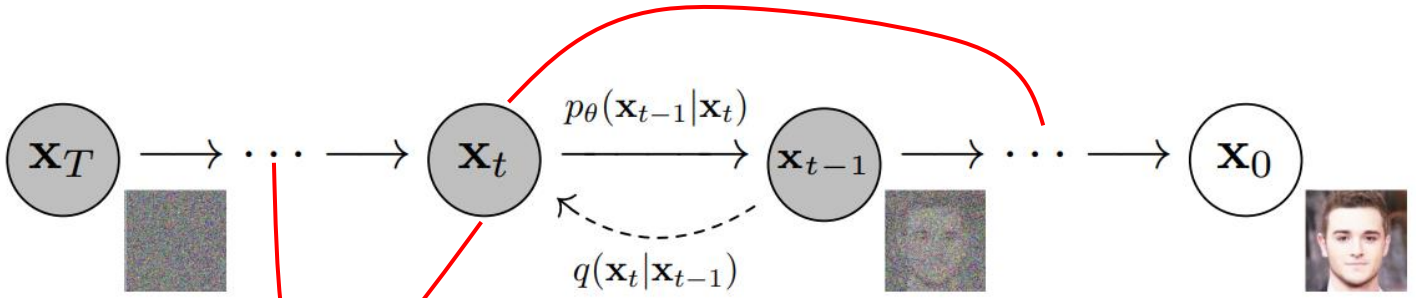


Figure 2: The directed graphical model considered in this work.

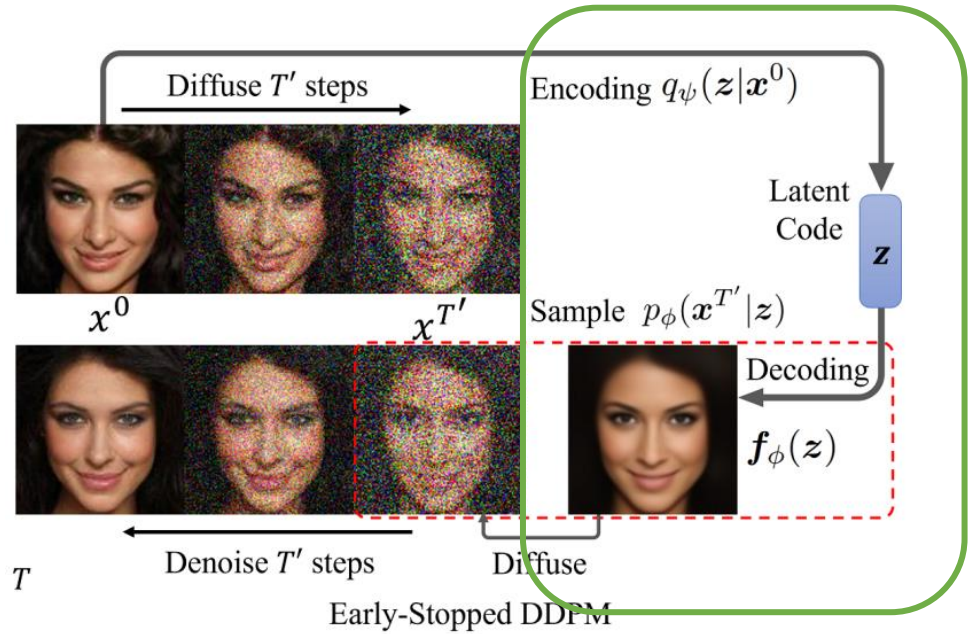
We can construct an ELBO path that entails $K \in \mathbb{N}$ refinement steps. I.e., for any K, and any given path of inference timesteps $0 = t'_0 < t'_1 < ... < t'_{K-1} < t'_K = 1$, one can derive a corresponding ELBO

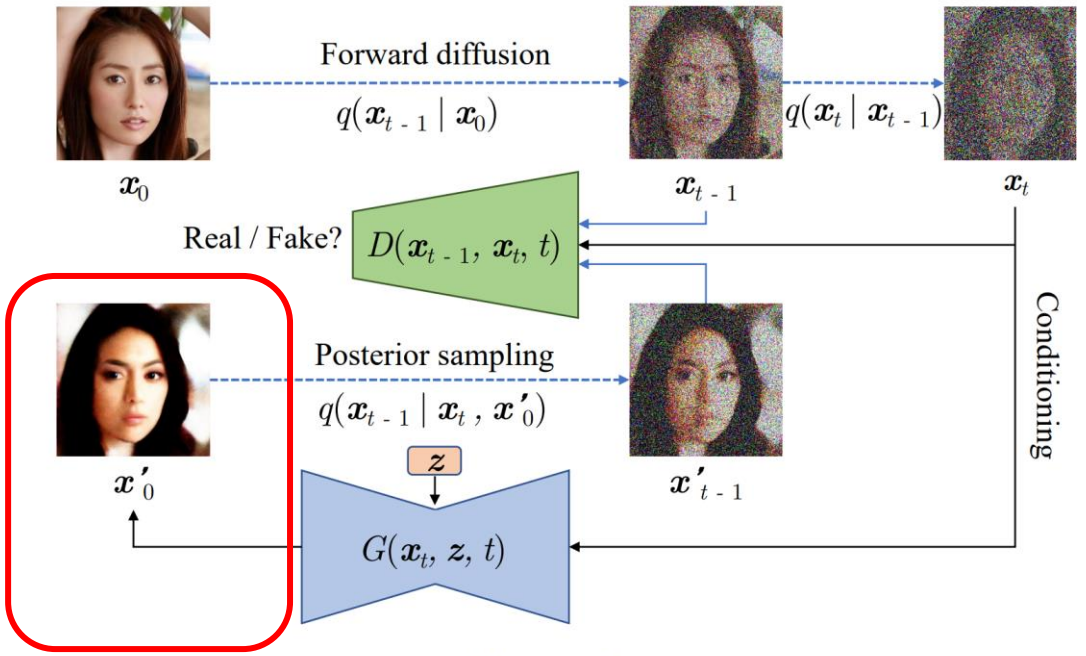
$$-L_{\text{ELBO}} = \mathbb{E}_q D_{\text{KL}}\big(q(\boldsymbol{x}_1|\boldsymbol{x}_0)\|p_\theta(\boldsymbol{x}_1)\big) + \sum_{i=1}^{K} L(t'_i, t'_{i-1}) \tag{16}$$

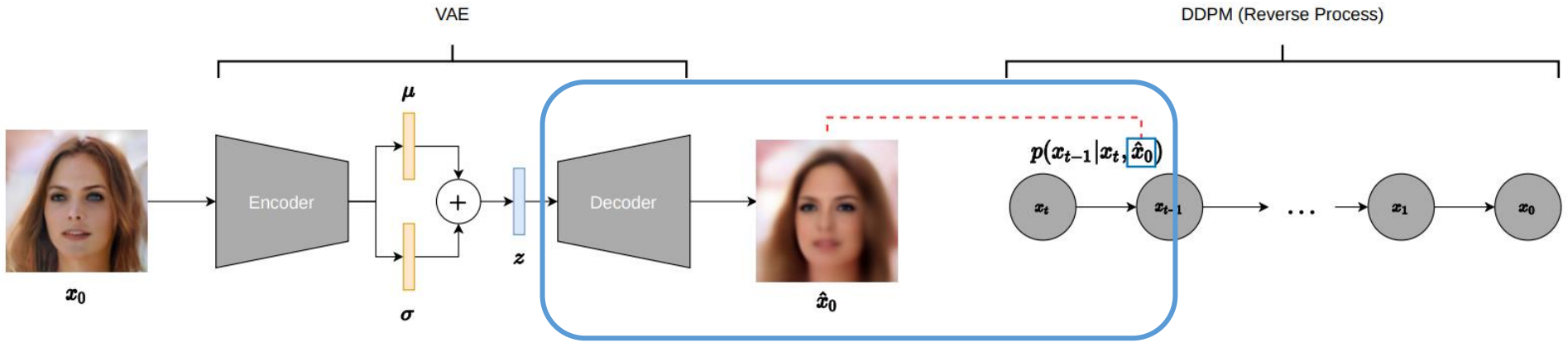
where

$$L(t,s) = \begin{cases} -\mathbb{E}_q \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_0) & s = 0 \\ \mathbb{E}_q D_{\text{KL}}\big(q(\boldsymbol{x}_s|\boldsymbol{x}_t,\boldsymbol{x}_0)\|p_\theta(\boldsymbol{x}_s|\boldsymbol{x}_t)\big) & s > 0 \end{cases} \tag{17}$$

Mixed-Modeling: Acceleration







Mixed-Modeling: Expressiveness

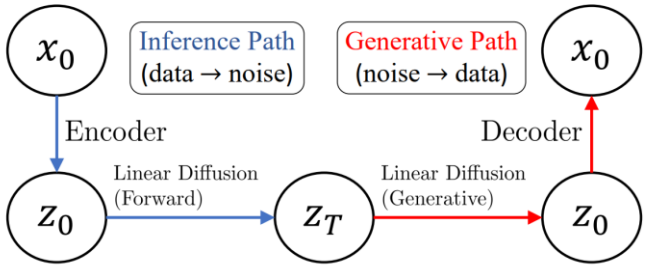


Figure 3: LSGM.

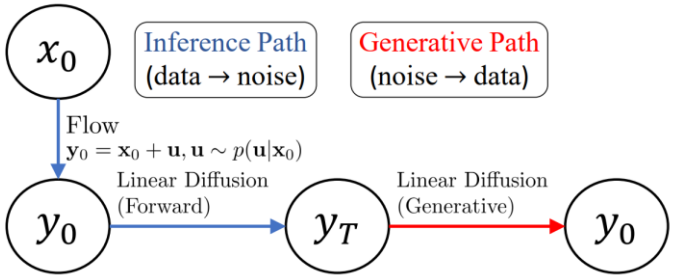


Figure 4: ScoreFlow.

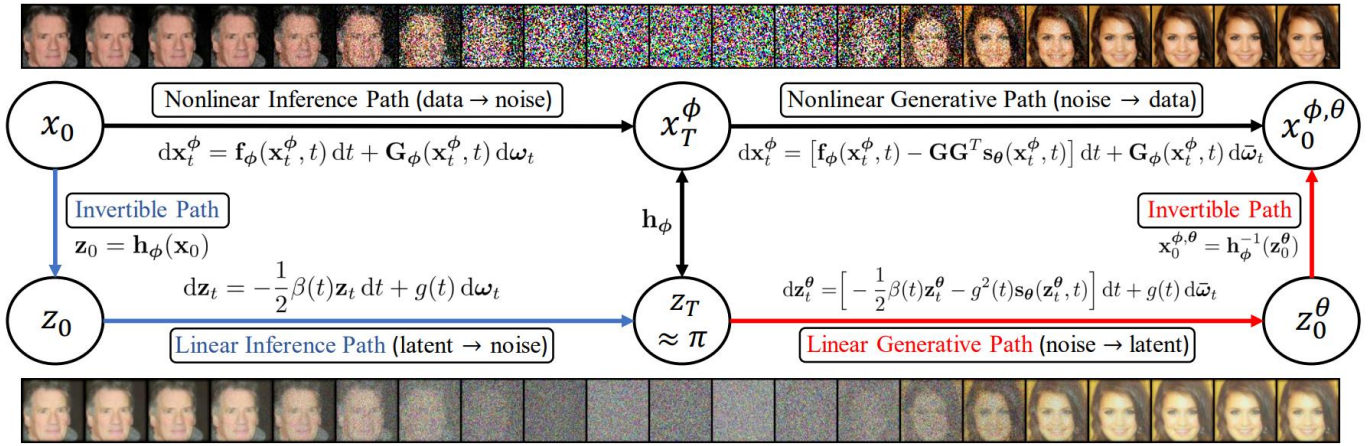
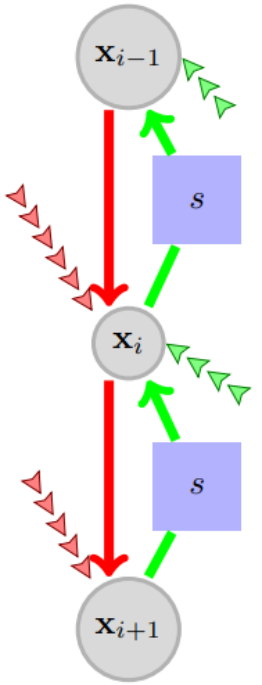
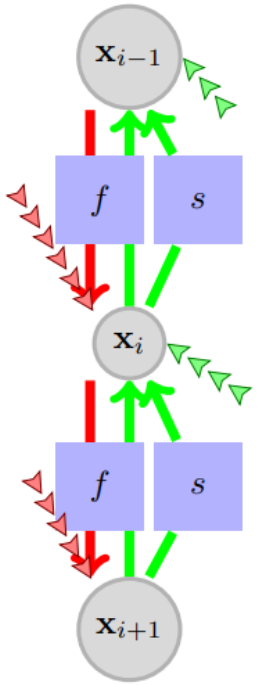


Figure 2: PDM attains a ladder structure between a data space and a latent space by the invertibility of normalizing flow. A score network estimates data score following blue arrows, and a model generates images following red arrows.



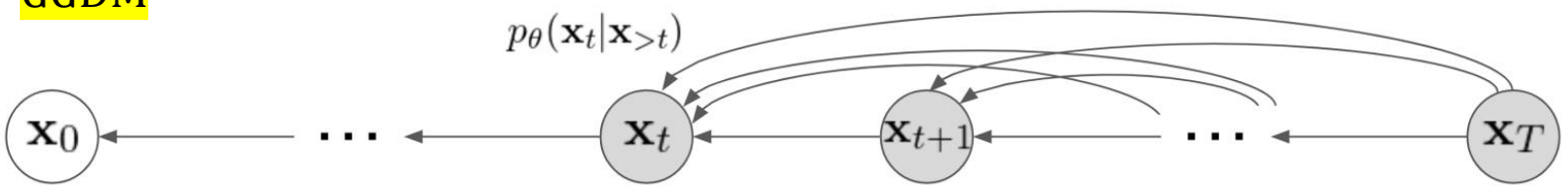
Diffusion Models



DiffFlow

Unification: Unified Framework

GGDM

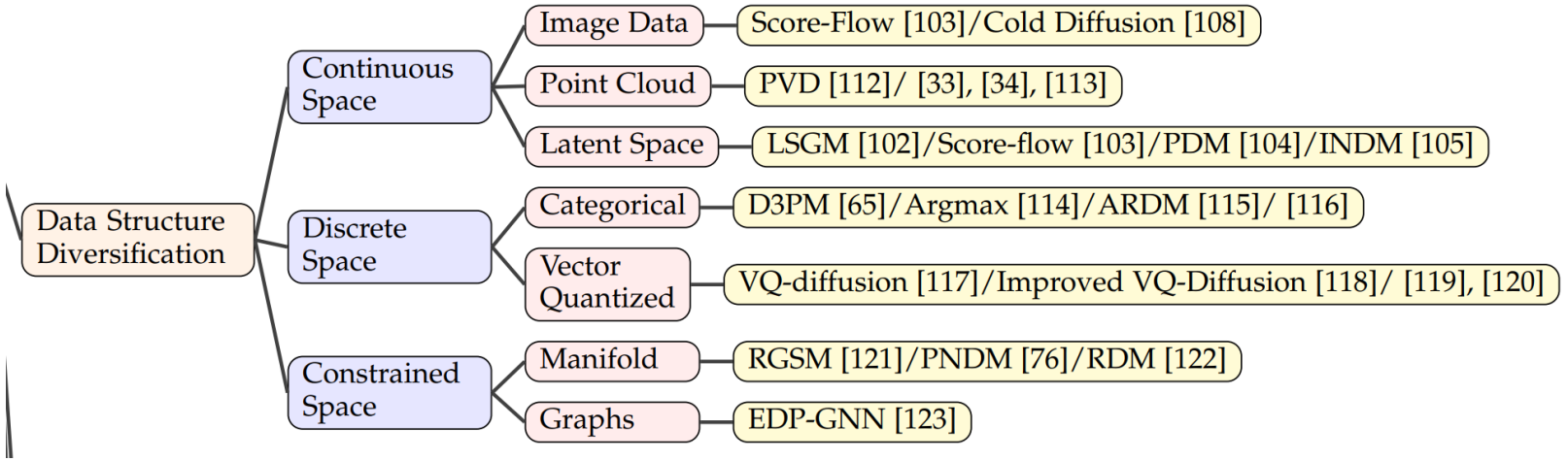


- ScoreSDE: Unified SDE Framework
- gDDIM: Generalized Implicit diffusion model
- VDM: Express the whole pipeline w.r.t. signal-to-noise ratio
- FastDPM: Express the whole pipeline w.r.t time-based function
- GGDM: Consider a fully-dependent diffusion framework
- Cold Diffusion: Summarize diffusion as Degrader and Reconstructor

Diffusion Needs Improvements

- Traditional RGB-image is not applicable for all fields.

1. Continuous Space
2. Discrete Space
3. Constrained Space



Diversification: Continuous

3D Point Cloud

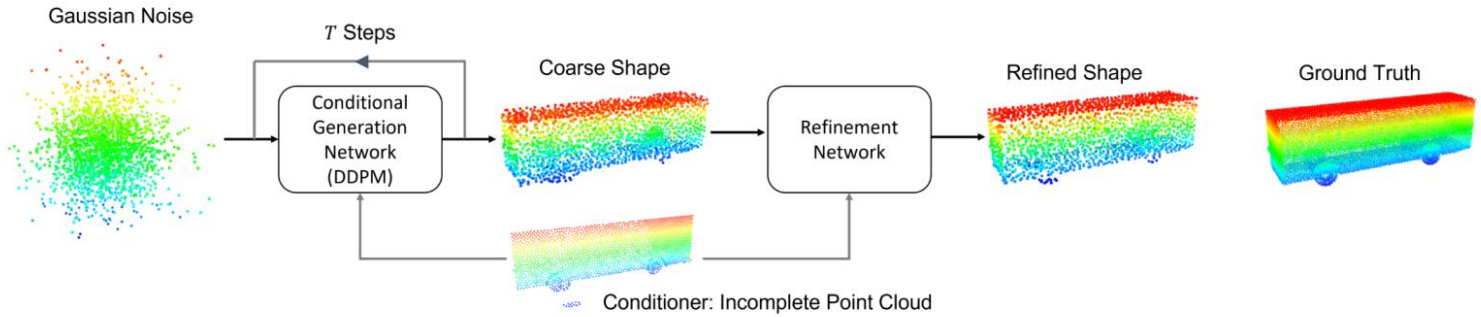
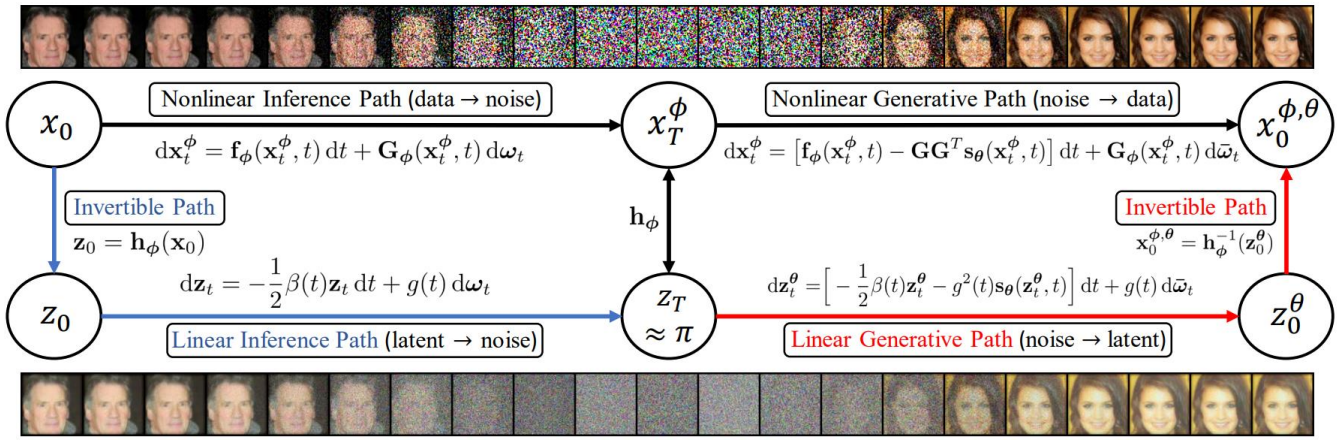


Figure 1: Our Conditional **P**oint **D**iffusion-**R**efinement (PDR) paradigm first moves a Gaussian noise step by step towards a coarse completion of the partial observation through a diffusion model (DDPM). Then it refines the coarse point cloud by one step to obtain a high quality point cloud.



Latent Space

Figure 2: PDM attains a ladder structure between a data space and a latent space by the invertibility of normalizing flow. A score network estimates data score following blue arrows, and a model generates images following red arrows.

Diversification: Discrete

Categorical/Multinomial

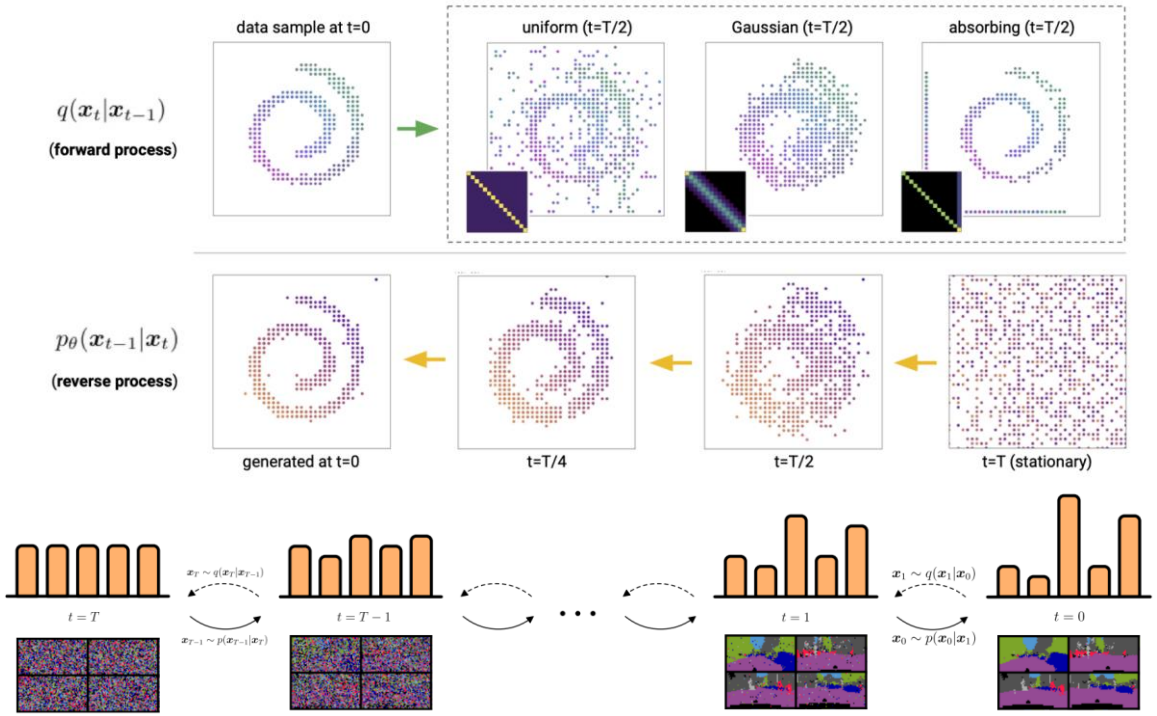


Figure 2: Overview of multinomial diffusion. A generative model $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ learns to gradually denoise a signal from left to right. An inference diffusion process $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ gradually adds noise form right to left.

Vector-Quantized

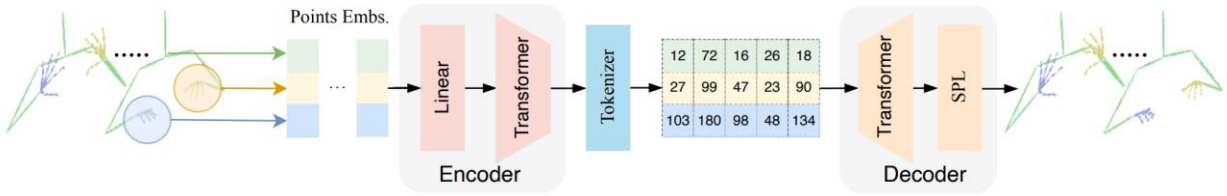
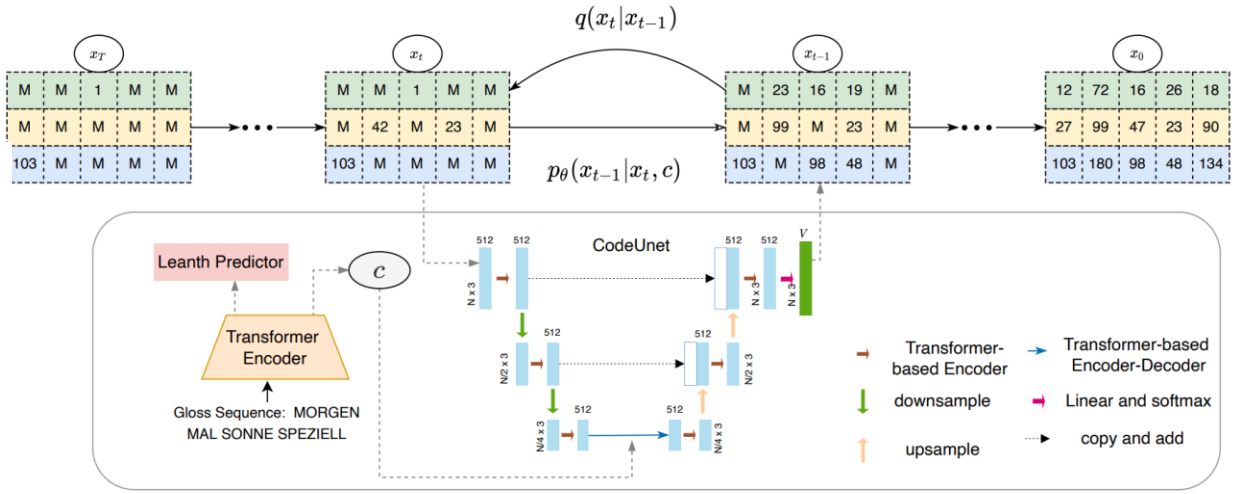
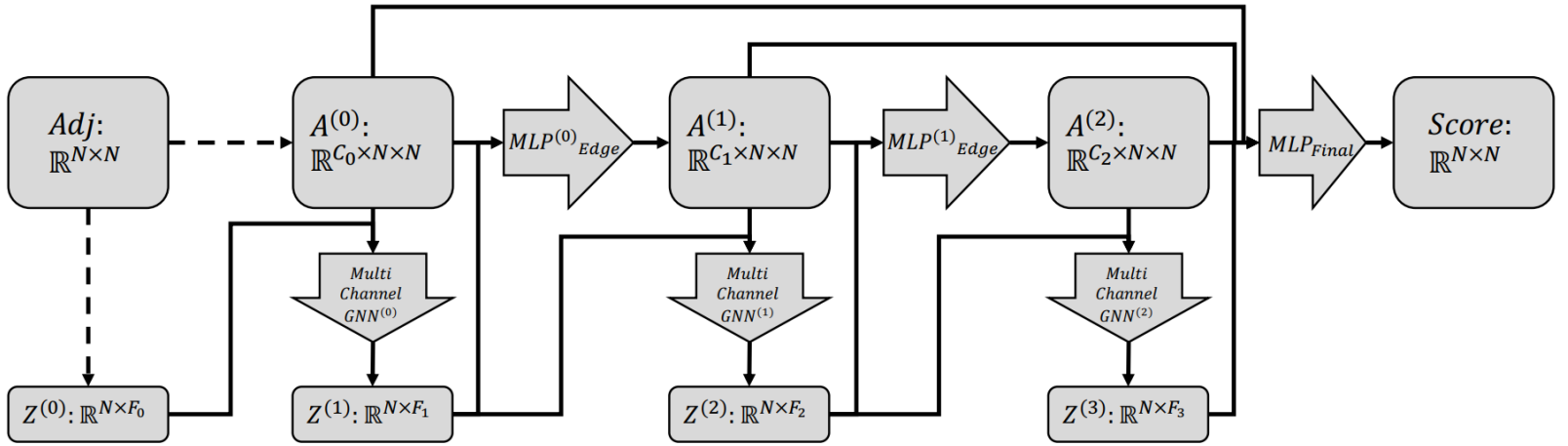


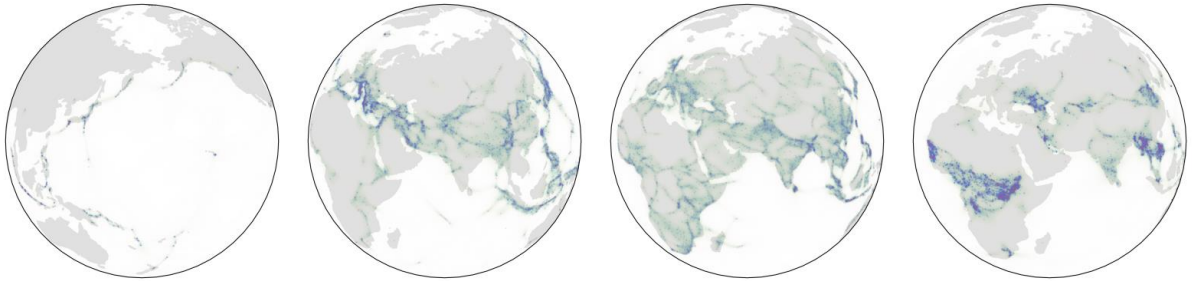
Figure 1: The architecture of the first stage model Pose-VQVAE for learning the discrete latent codes.



Diversification: Constrained

Graph





(a) Volcano (b) Earthquake (c) Flood (d) Fire

Figure 1: Trained score-based generative models on earth sciences data. The learned density is colored green-blue. Blue and red dots represent training and testing datapoints, respectively.

Manifold

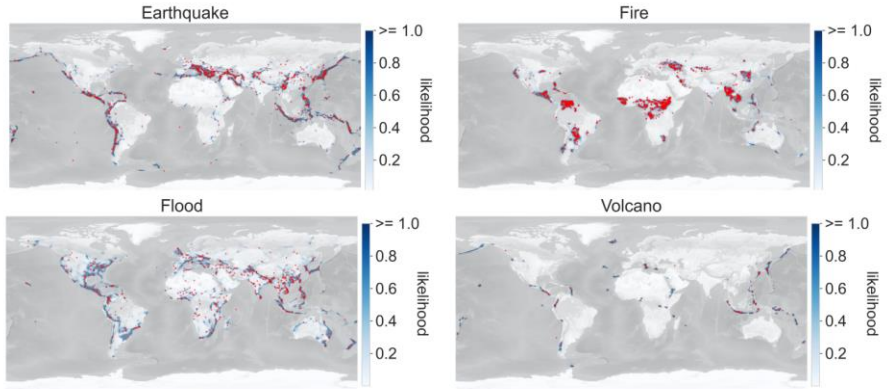
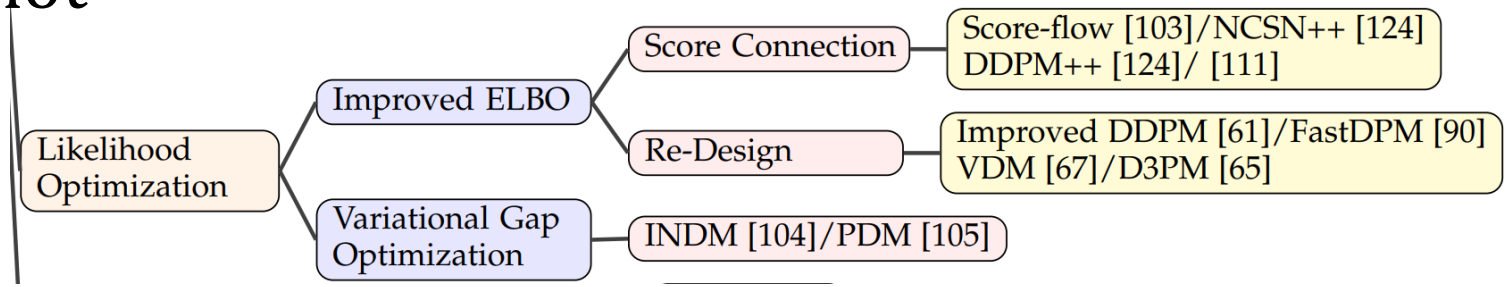


Figure 1: Density of models trained on earth datasets. Red dots are samples from the test set.

Diffusion Needs Improvements

- Likelihood Optimization achieved by ELBO may not work sometimes.



1. Advanced ELBO Design
2. Variational Gap Optimization

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*

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

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

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

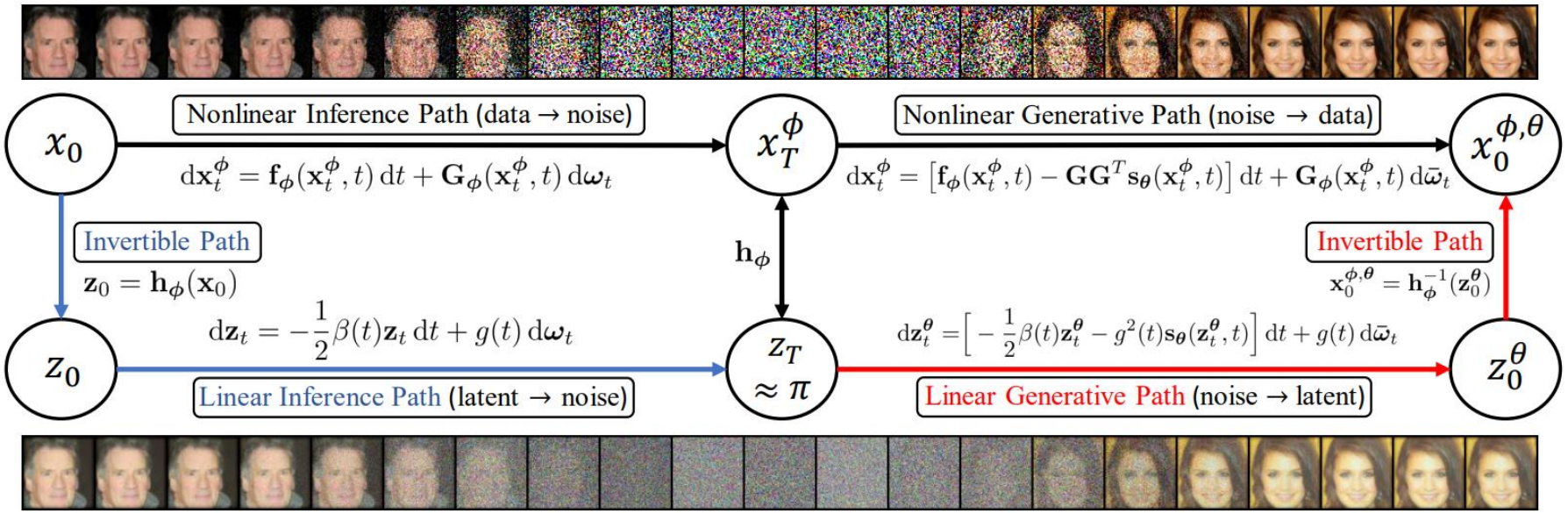
$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{vlb}}$$

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

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

Re-Design: design vlb loss from a different perspective to obtain better convergence

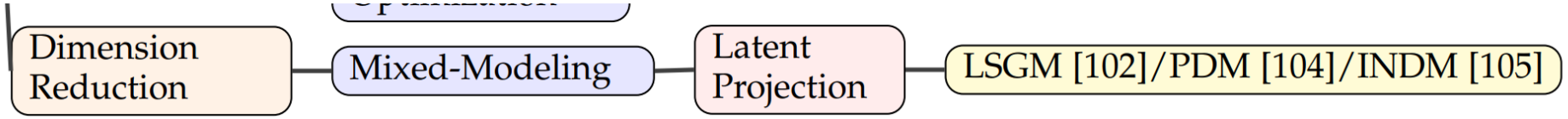
Likelihood Optimization: Variational Gap



$$\begin{aligned}\mathrm{Gap}_d(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d, \boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) =& \mathrm{NELBO} - \mathrm{NLL} \\ =& D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d \| \boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) - D_{KL}(p_r \| p_{\boldsymbol{\phi},\boldsymbol{\theta}}),\end{aligned}$$

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathrm{Gap}_d(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d, \boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) = \arg\min_{\boldsymbol{\phi}} D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d \| \boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) = \arg\min_{\boldsymbol{\phi}} D_{KL}(p_r \| p_{\boldsymbol{\phi},\boldsymbol{\theta}}).$$

Diffusion Needs Improvements



- Gradually transferred latent space cannot reduce the dimension

1. NF/VAE will help you

Dimension Reduction: Latent Projection

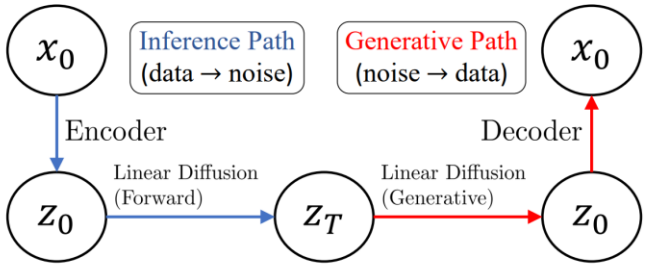


Figure 3: LSGM.

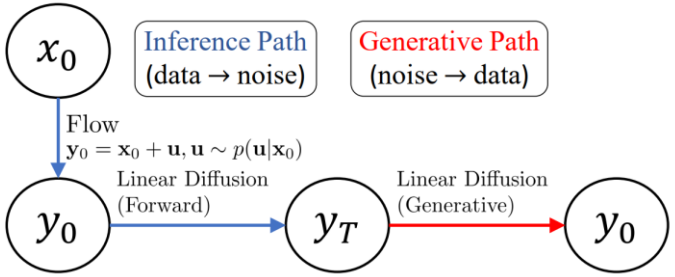


Figure 4: ScoreFlow.

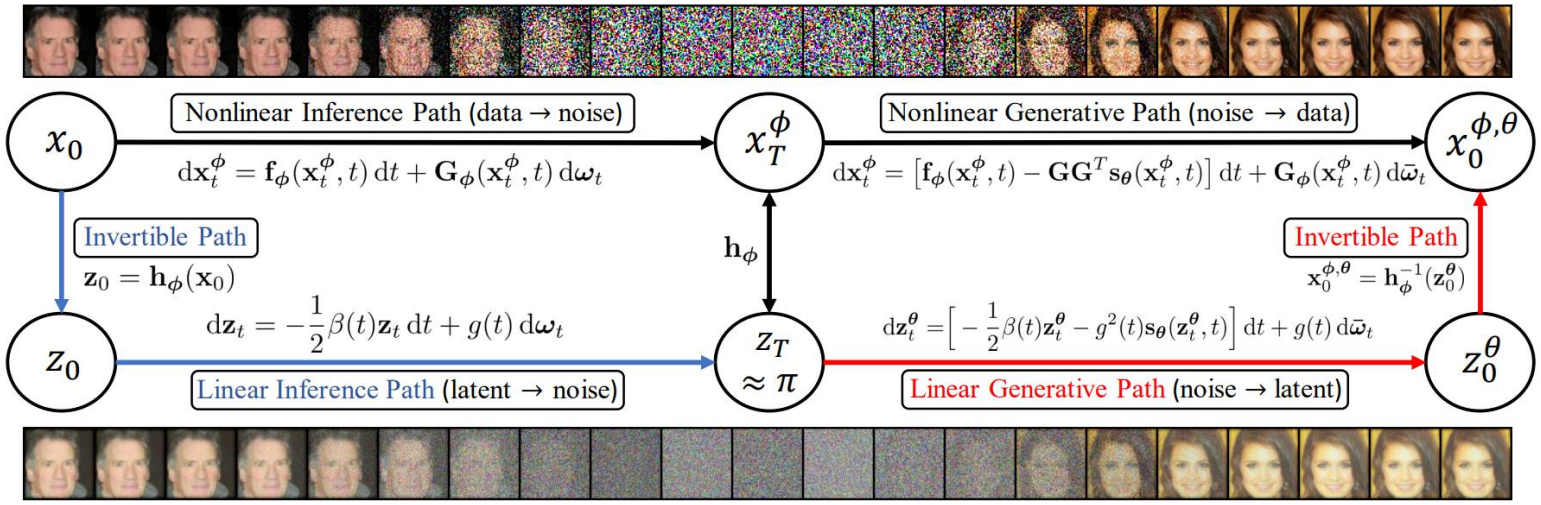
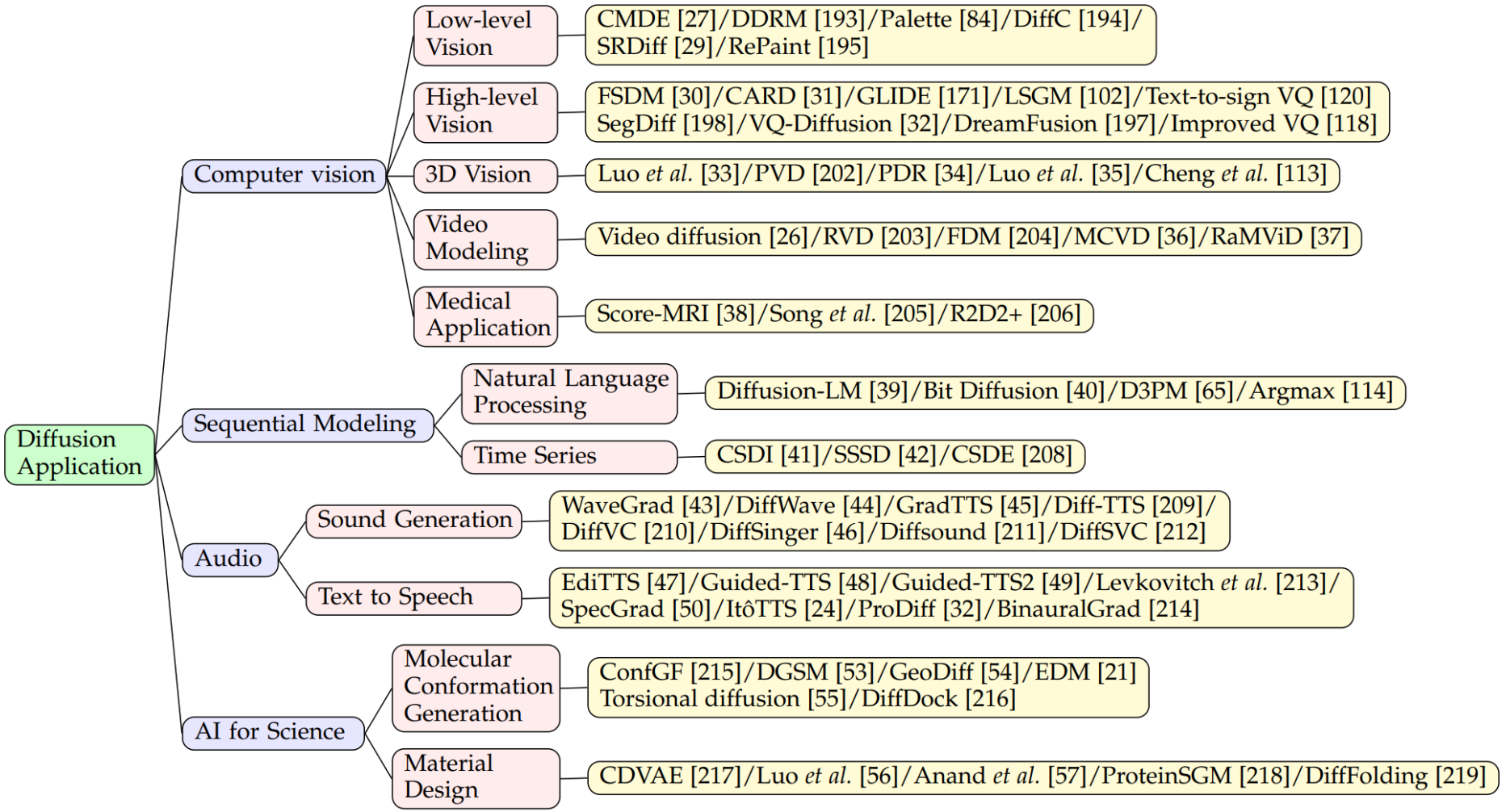


Figure 2: PDM attains a ladder structure between a data space and a latent space by the invertibility of normalizing flow. A score network estimates data score following blue arrows, and a model generates images following red arrows.

Diffusion Has Applications



Low-Level Vision:

CMDE [27]	2021	RGB-Image	SDE	Inpainting, Super-Resolution, Edge to image translation
DDRM [193]	2022	RGB-Image	Diffusion	Super-Resolution, Deblurring, Inpainting, Colorization
Palette [84]	2022	RGB-Image	Diffusion	Colorization, Inpainting, Uncropping, JPEG Restoration
DiffC [194]	2022	RGB-Image	SDE	Compression
SRDiff [29]	2021	RGB-Image	Diffusion	Super-Resolution -
RePaint [195]	2022	RGB-Image	Diffusion	Inpainting, Super-resolution, Edge to Image Translation

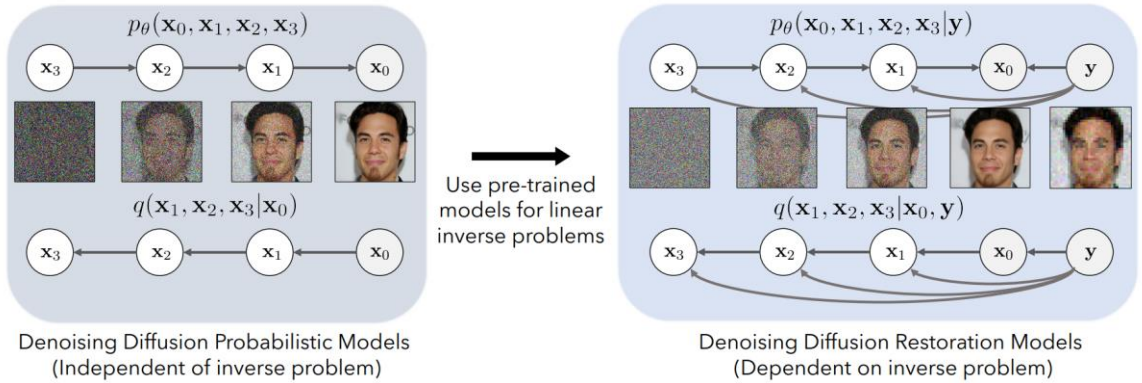
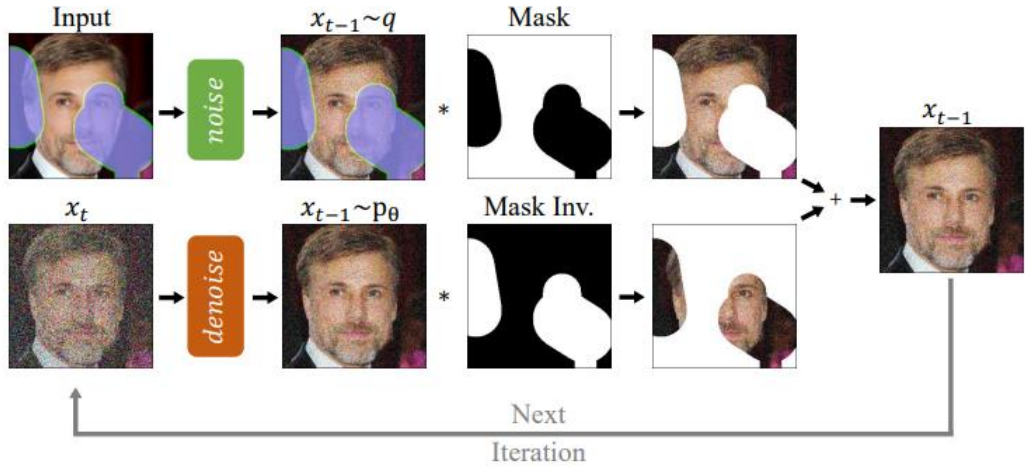
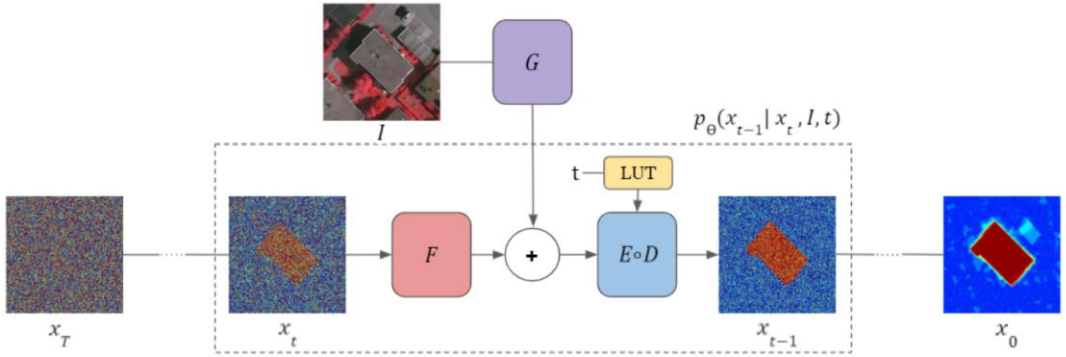


Figure 2. Illustration of our DDRM method for a specific inverse problem (super-resolution + denoising). We can use unsupervised DDPM models as a good solution to the DDRM objective.



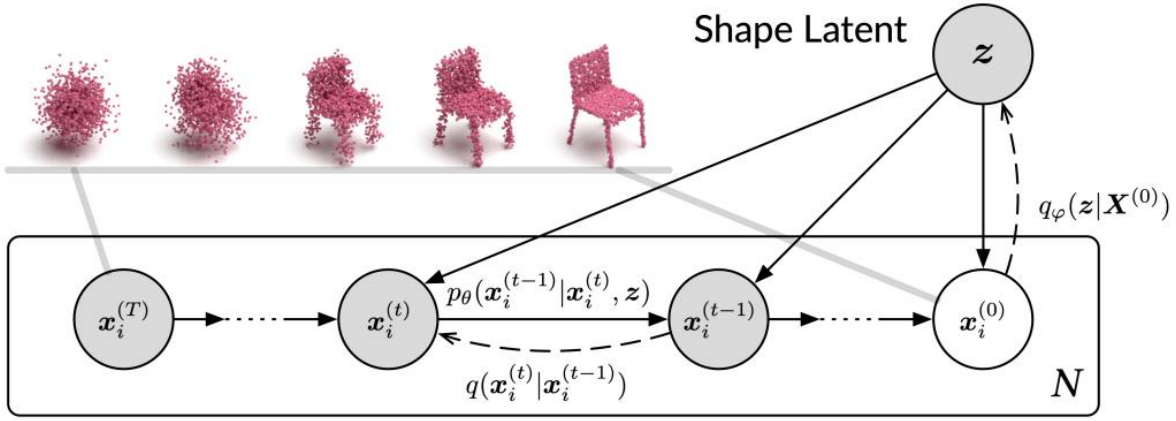
High-Level Vision:

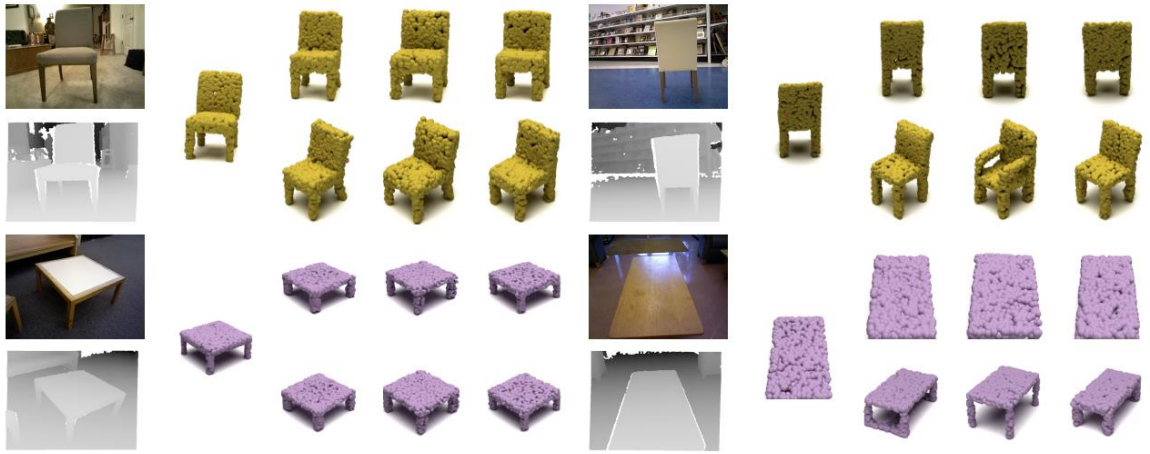
FSDM [30]	2022	RGB-Image	Diffusion	Few-shot Generation
CARD [31]	2022	RGB-Image	Diffusion	Conditional Generation
GLIDE [171]	2022	RGB-Image	Diffusion	Conditional Generation
LSGM [102]	2022	RGB-Image	SDE	UnConditional & Conditional Generation
SegDiff [198]	2022	RGB-Image	Diffusion	Segmentation
VQ-Diffusion [117]	2022	VQ Data	Diffusion	Text-to-Image Synthesis
DreamFusion [197]	2023	VQ Data	Diffusion	Text-to-Image Synthesis
Text-to-Sign VQ [120]	2022	VQ Data	Diffusion	Conditional Pose Generation
Improved VQ-Diff [118]	2022	VQ Data	Diffusion	Text-to-Image Synthesis



3D Vision:

Luo's Model [33]	2021	Point Cloud	Diffusion	Point Cloud Generation
PVD [202]	2022	Point Cloud	Diffusion	Point Cloud Generation, Point-Voxel representation
PDR [34]	2022	Point Cloud	Diffusion	Point Cloud Completion
Cheng's Model [113]	2022	Point Cloud	Diffusion	Point Cloud Generation
Luo's Model [35]	2022	Point Cloud	Score	Point Cloud Denoising





Video Modeling:

VDM [26]	2022	Video	Diffusion	Text-Conditioned Video Generation
RVD [203]	2022	Video	Diffusion	Video Forecasting, Video compression
FDM [204]	2022	Video	Diffusion	Video Forecasting, Long-range Video modeling
MCVD [36]	2022	Video	Diffusion	Video Prediction, Video Generation, Video Interpolation
RaMViD [37]	2022	Video	SDE	Conditional Generation

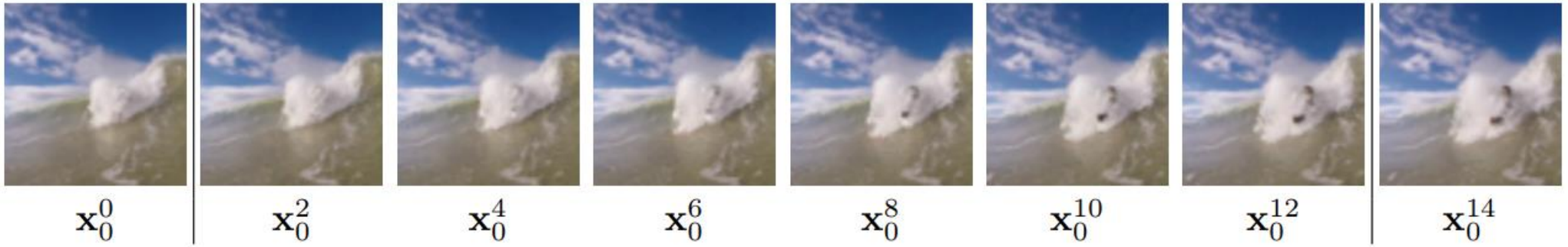
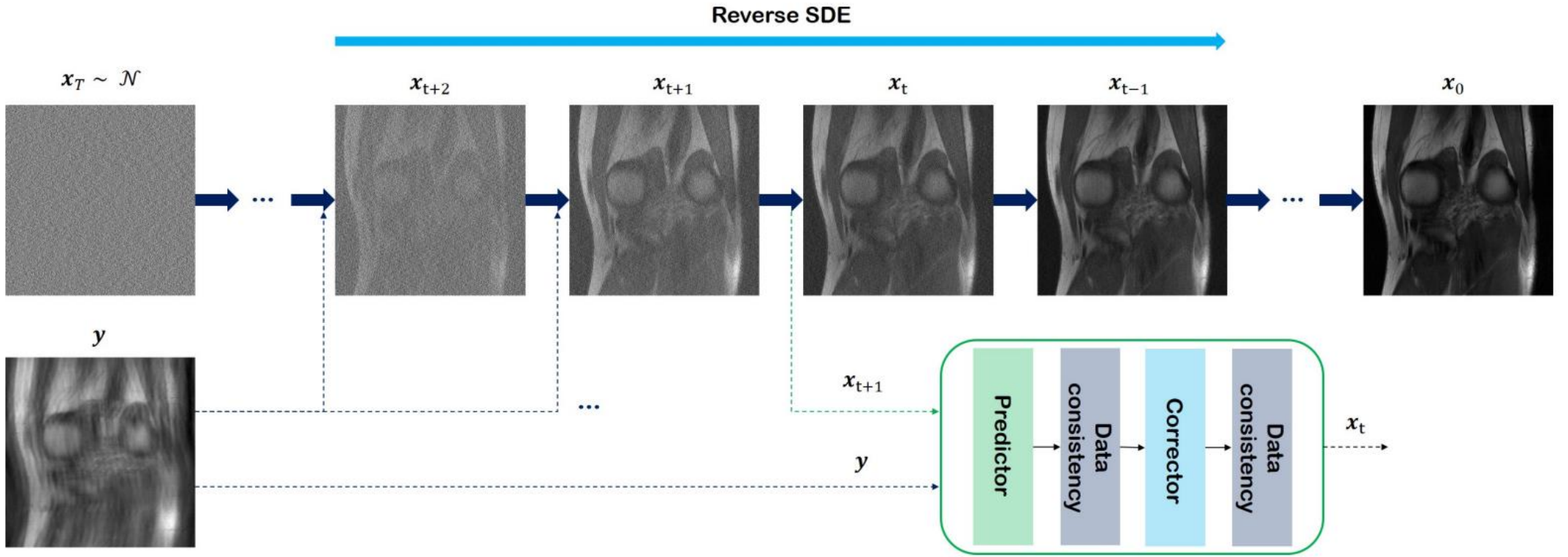


Figure 1: The first two and last two frames of a video are given and our model does fill in the missing frames very accurate and with much detail.

Medical Image Processing:

Score-MRI [38]	2022	MRI	SDE	MRI Reconstruction
Song's Model [205]	2022	MRI, CT	SDE	MRI Reconstruction, CT Reconstruction
R2D2+ [206]	2022	MRI	SDE	MRI Denoising



Sequence Modeling: NLP

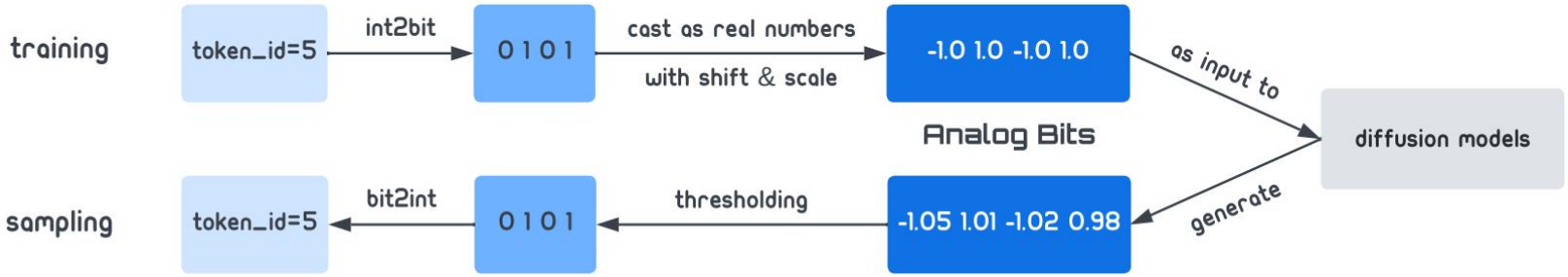
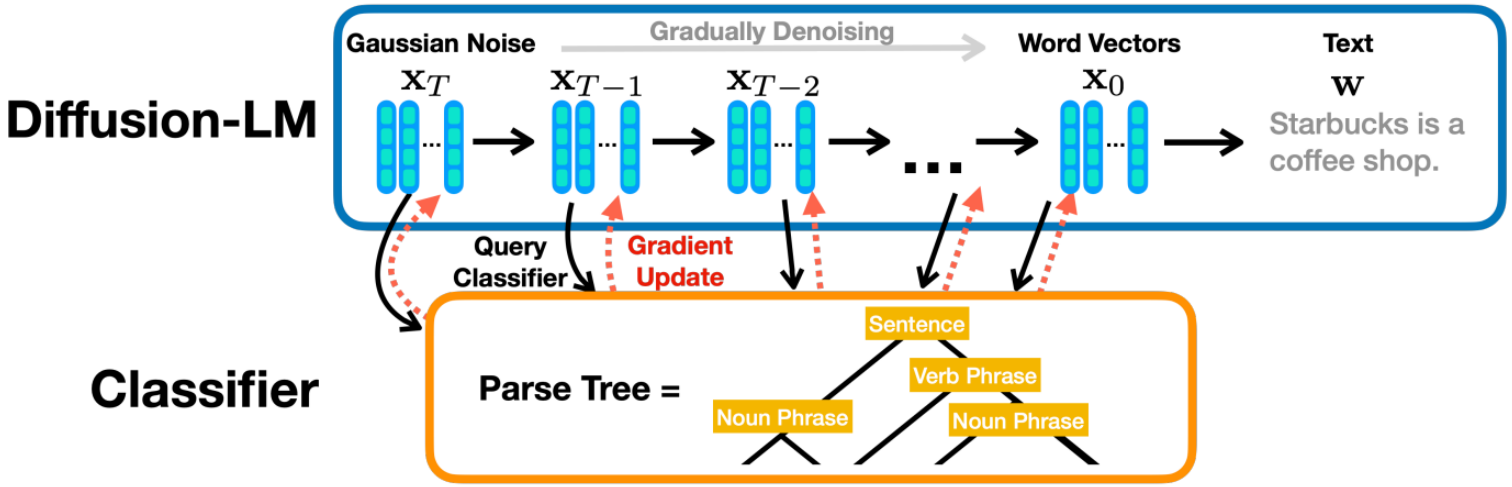
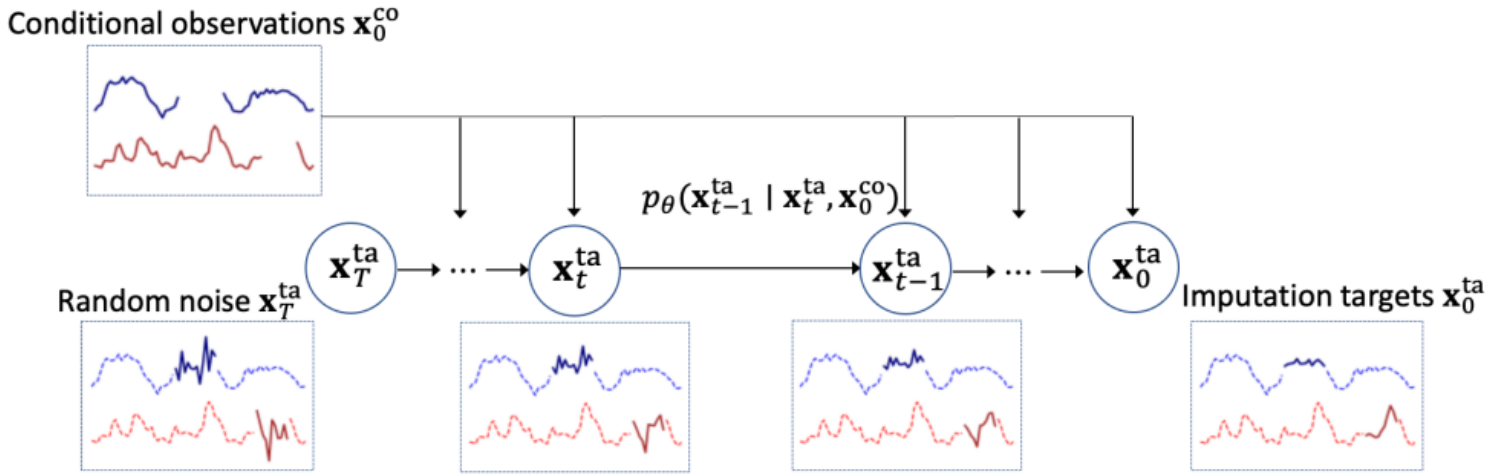


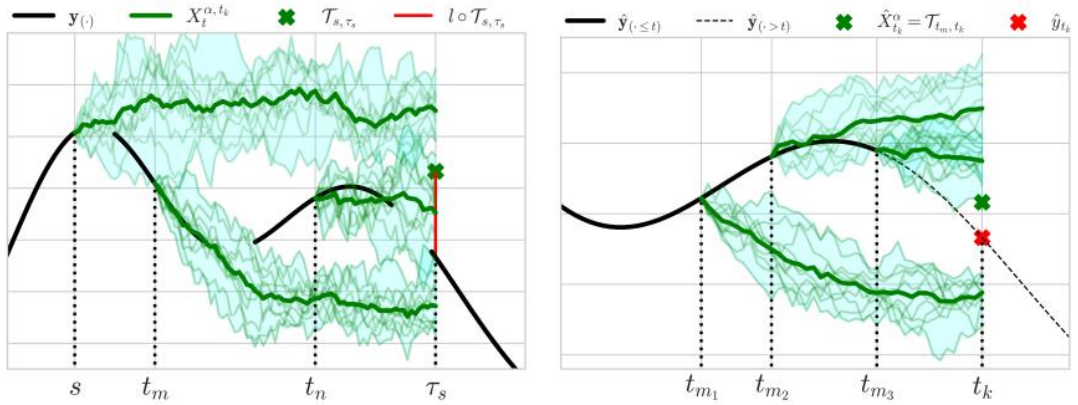
Figure 1: Bit Diffusion: modeling discrete data using continuous diffusion models with analog bits.



Sequence Modeling: Time Series



Conditioning matters!!!



(a) Network Training with the MFcond loss

(b) Network Inference

Audio: Sound Generation

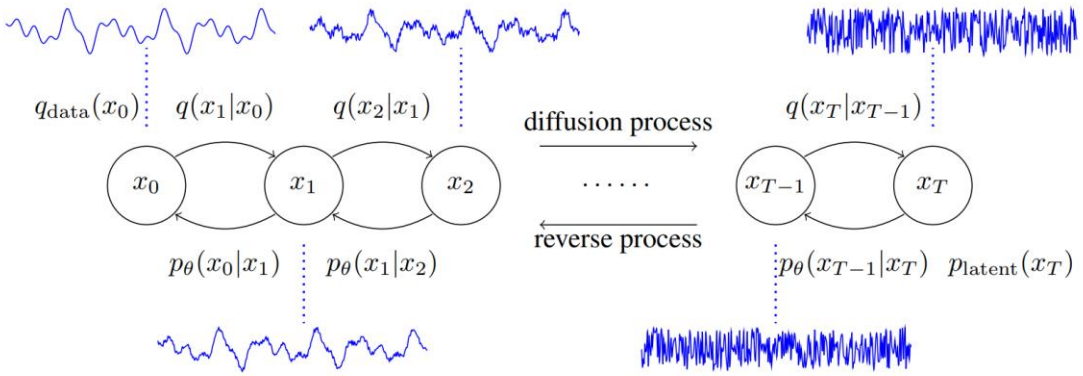


Figure 1: The diffusion and reverse process in diffusion probabilistic models. The reverse process gradually converts the white noise signal into speech waveform through a Markov chain $p_\theta(x_{t-1}|x_t)$.

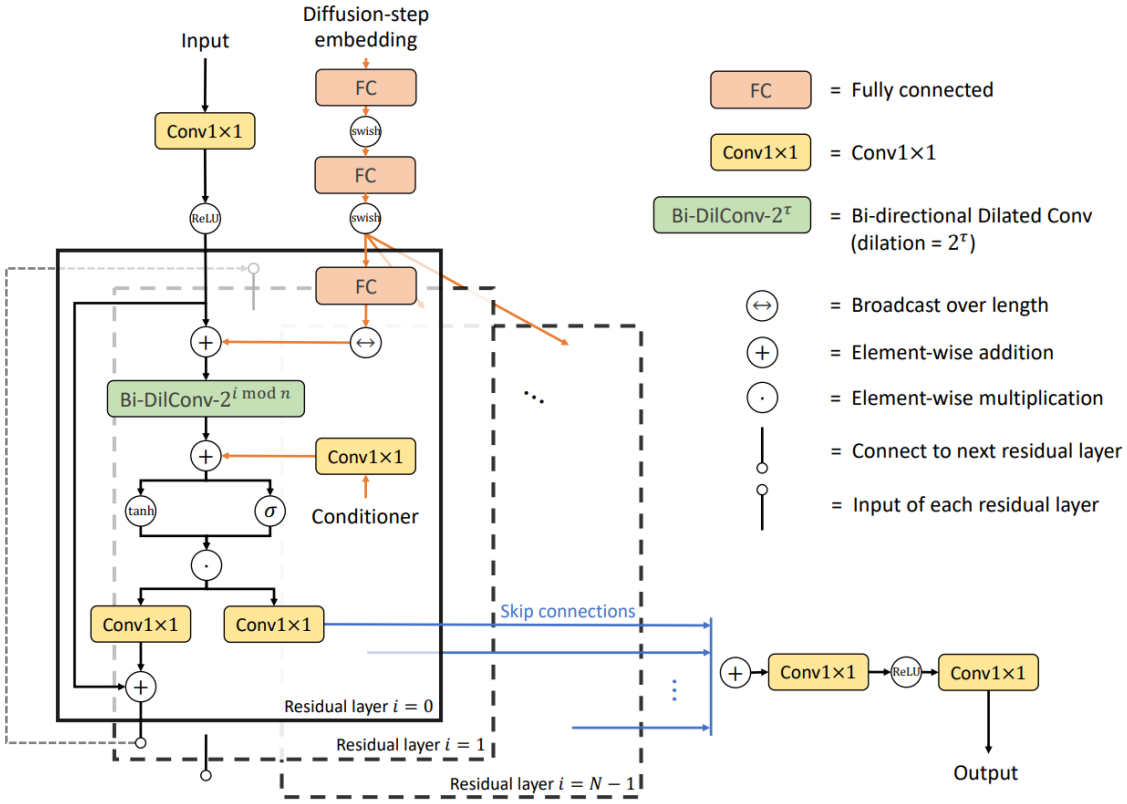


Figure 2: The network architecture of DiffWave in modeling $\epsilon_\theta : \mathbb{R}^L \times \mathbb{N} \to \mathbb{R}^L$.

Audio: Text to Speech

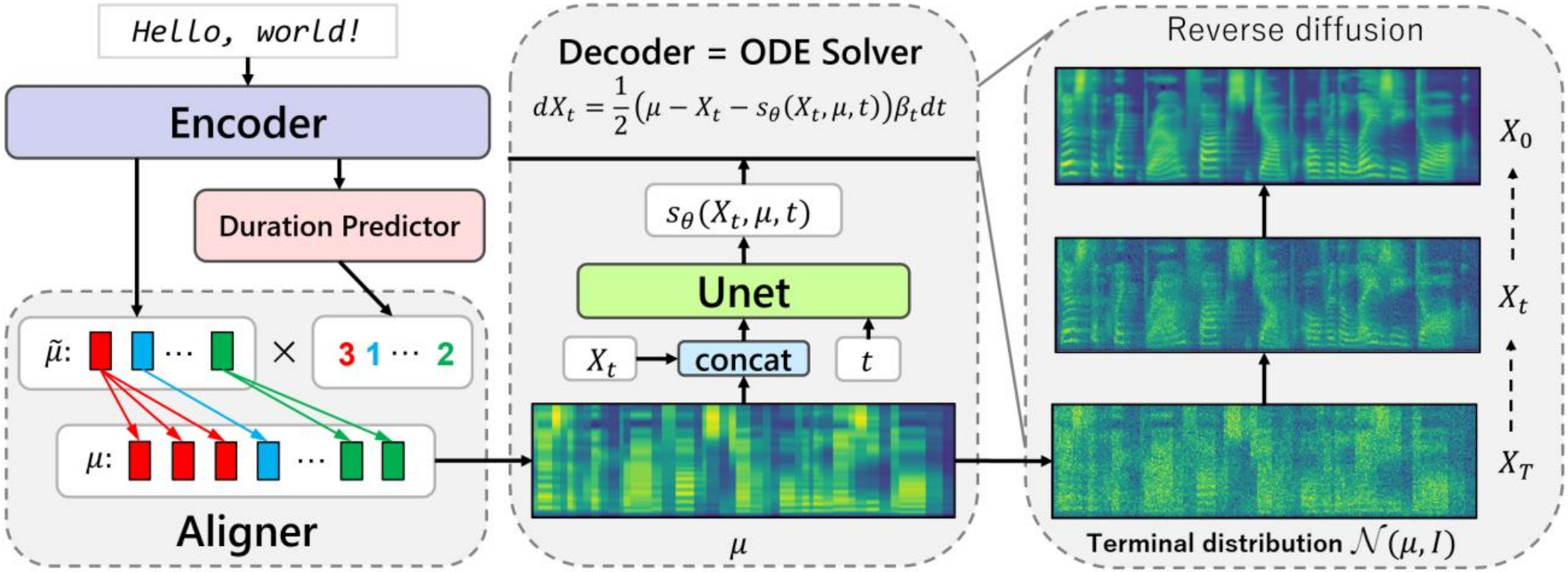
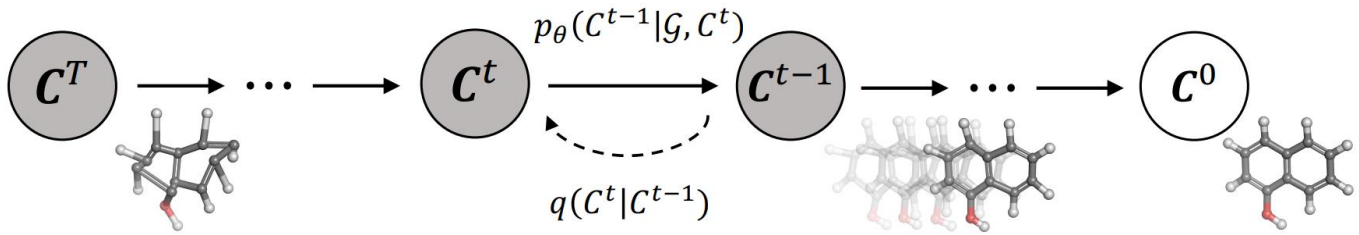
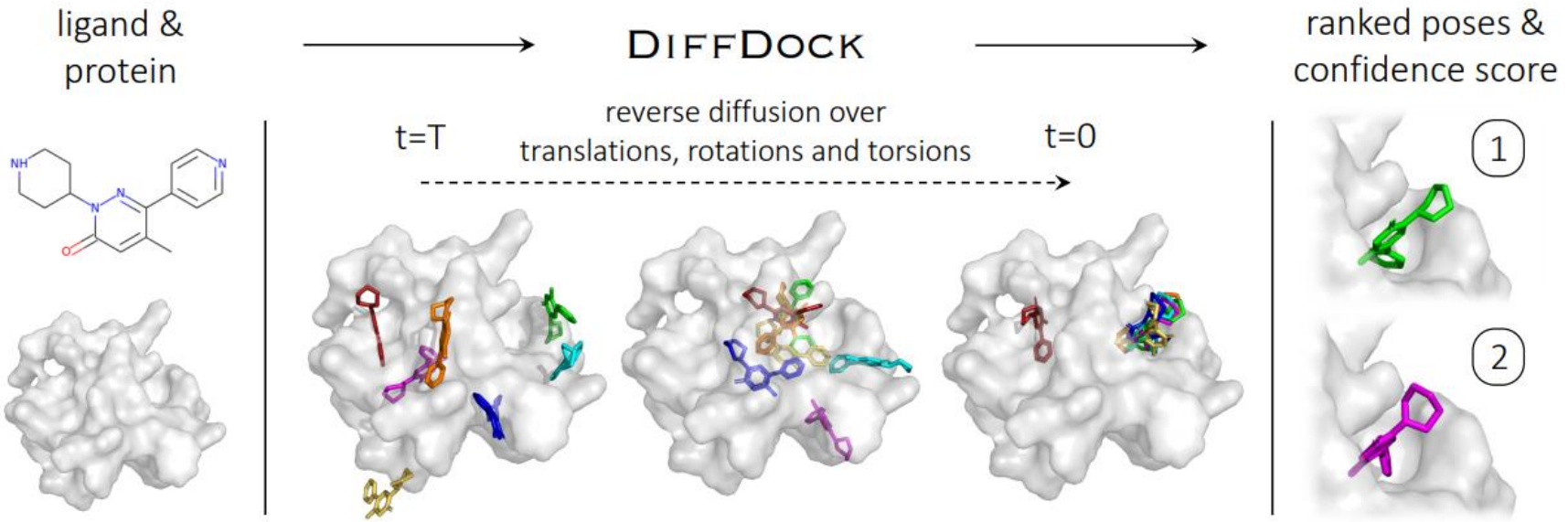


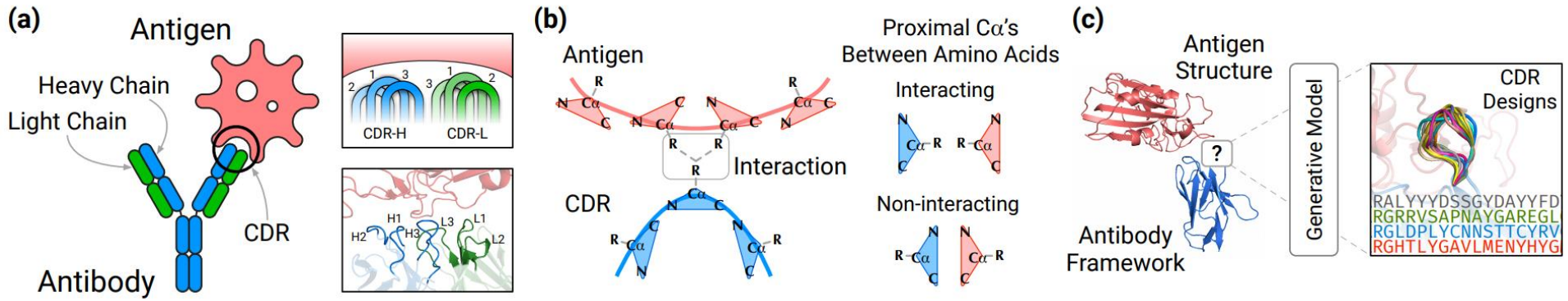
Figure 2. Grad-TTS inference scheme.

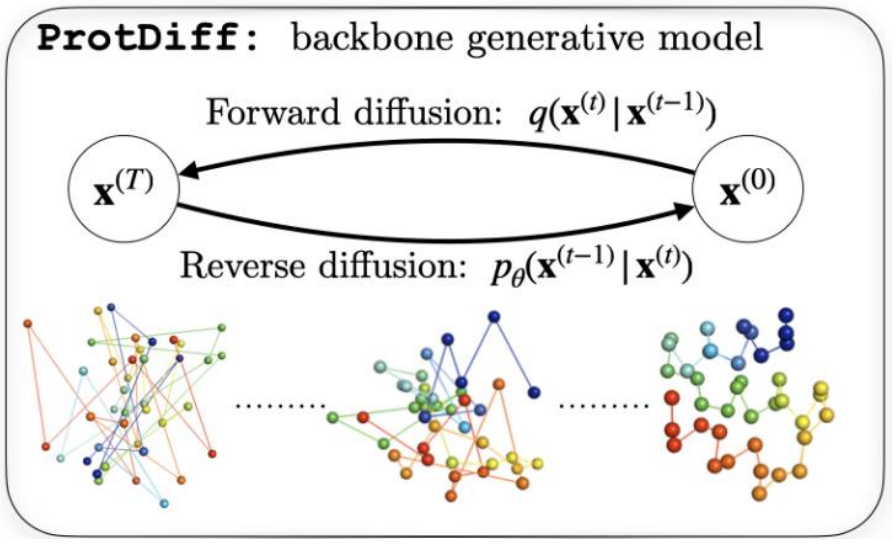
AI for Science: Molecular Generation

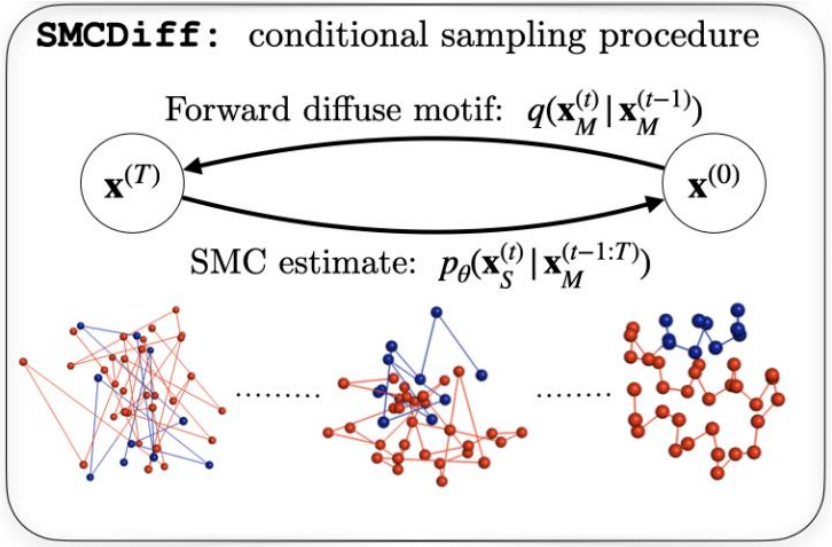




AI for Science: Protein Modeling







Further Directions:

- 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

Materials: 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!