DDPM

To generate new fake images being the same kind as the input images.

Forward: ① Given a ground-truth image to be learned to output

a mean to ensure feature closeness between generated & ground-truth

each step's output provides a snapshot for the reverse process to learn

② Given a designed form to add steps of noises while each noise is randomly sampled from a decide distribution

$$X_t = \mu_t + \sigma_t \odot \epsilon,$$
$$= \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \epsilon,$$
$$\epsilon \sim N(0, I)$$

a designed form

a randomly sampled noise (from an also decided distribution)

a mean to introduce stochasticity to later model of the reverse process

③ Get pure-noises image

(meaning the entire map is of the decided distribution (structural info. is completely lost, has no statistical difference compared to any other pure-noise map of the same distribution) → a standardized form of the starting state of the reverse process

Reverse: ① Start from the pure-noises image

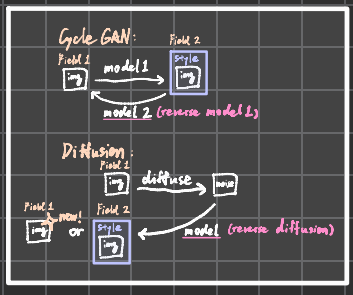
② In each step:

a. use x_t (to represent) & $\overline{\epsilon^{(t)}}$ (the ground-truth) known from `Forward` to learn $\epsilon_\theta(x_t, t)$ and get the distribution of x_{t-1}

b. sample a $\tilde{x}_{t-1}$

a mean to introduce stochasticity to later model of the reverse process

③ Reach the distribution of x_0 in the end then sample any new image



Forward : diffusion (inferrence)

Assume that: $p(x_t|x_0, x_1, \dots, x_{t-1}) = p(x_t|x_{t-1})$

Designed → ① Markov Chain : of ② $q(x_t|x_{t-1}) = N(x_t; \underbrace{\sqrt{1-\beta_t}\cdot x_{t-1}}_{\mu}, \underbrace{\beta_t I}_{\sigma^2})$

where ③ $\beta_t = S(t)$ (a Variance Schedule <diffusion rate> that $t\uparrow \Rightarrow \beta_t\uparrow$)

Use sinusoidal position embedding

$$\Rightarrow q(x_{1:T}|x_0) = \frac{q(x_{0:T})}{q(x_0)} = \frac{q(x_{0:t-1})\cdot q(x_t|x_{0:t-1})}{q(x_0)} = \frac{q(x_{0:t-2})\cdot q(x_{t-1}|x_{0:t-2})\cdot q(x_t|x_{0:t-1})}{q(x_0)} = \dots$$

$$= \frac{q(x_0,x_1)\cdot q(x_2|x_0,x_1)\cdot \dots \cdot q(x_T|x_{0:T-1})}{q(x_0)} = \frac{q(x_0)\cdot q(x_1|x_0)\cdot \dots \cdot q(x_T|x_{0:T-1})}{q(x_0)} \quad \text{(Markov Chain Assumption)} \quad \frac{q(x_0)\cdot q(x_1|x_0)\cdot q(x_2|x_1)\cdot \dots \cdot q(x_T|x_{T-1})}{q(x_0)}$$

$$= \prod_{t=1}^{T} q(x_t|x_{t-1}) = \prod_{t=1}^{T} N(x_t; \sqrt{1-\beta_t}\cdot x_{t-1}, \beta_t I) \Rightarrow \text{the Forward Proc. (Diffusion)}$$

Use the Multi-parameter Method instead of the original N to be differentiable.

$$x_t = \sqrt{1-\beta_t}\cdot x_{t-1} + \sqrt{\beta_t}\cdot \epsilon^{(1)} \quad (\epsilon^{(1)}: \text{Sample once from } N(0,I))$$

$$= \sqrt{1-\beta_t}\left(\sqrt{1-\beta_{t-1}}\cdot x_{t-2} + \sqrt{\beta_{t-1}}\cdot \bar{\epsilon}^{(2)}\right) + \sqrt{\beta_t}\cdot \epsilon^{(1)} \quad (\bar{\epsilon}^{(2)}: \text{Sample twice from } N(0,I))$$

$$= \sqrt{(1-\beta_t)\cdot(1-\beta_{t-1})}\cdot x_{t-2} + \sqrt{(1-\beta_t)\beta_{t-1}}\cdot \epsilon^{(2)} + \sqrt{\beta_t}\cdot \epsilon^{(1)}$$

$$= \sqrt{(1-\beta_t)\cdot(1-\beta_{t-1})}\cdot x_{t-2} + \sqrt{1-(1-\beta_t)(1-\beta_{t-1})}\cdot \epsilon^{(2)} \quad (\Leftrightarrow \text{Embed noise from } \alpha_t = 1-\beta_t \text{ } N(0,I) \text{ twice})$$

$$= \sqrt{\alpha_t\cdot(1-\alpha_{t-1})}\cdot x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\cdot \epsilon^{(2)}$$

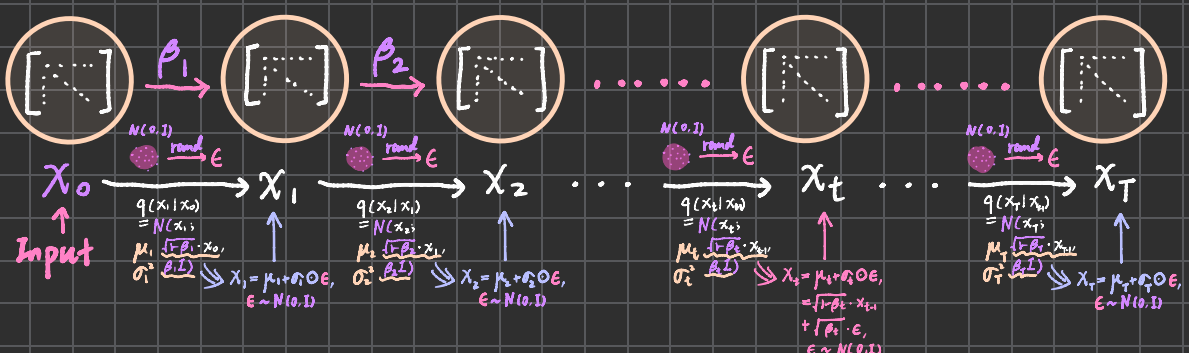
$$= \cdots = \sqrt{\prod_{i=1}^{t}\alpha_i}\cdot x_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\cdot \epsilon^{(t)}$$

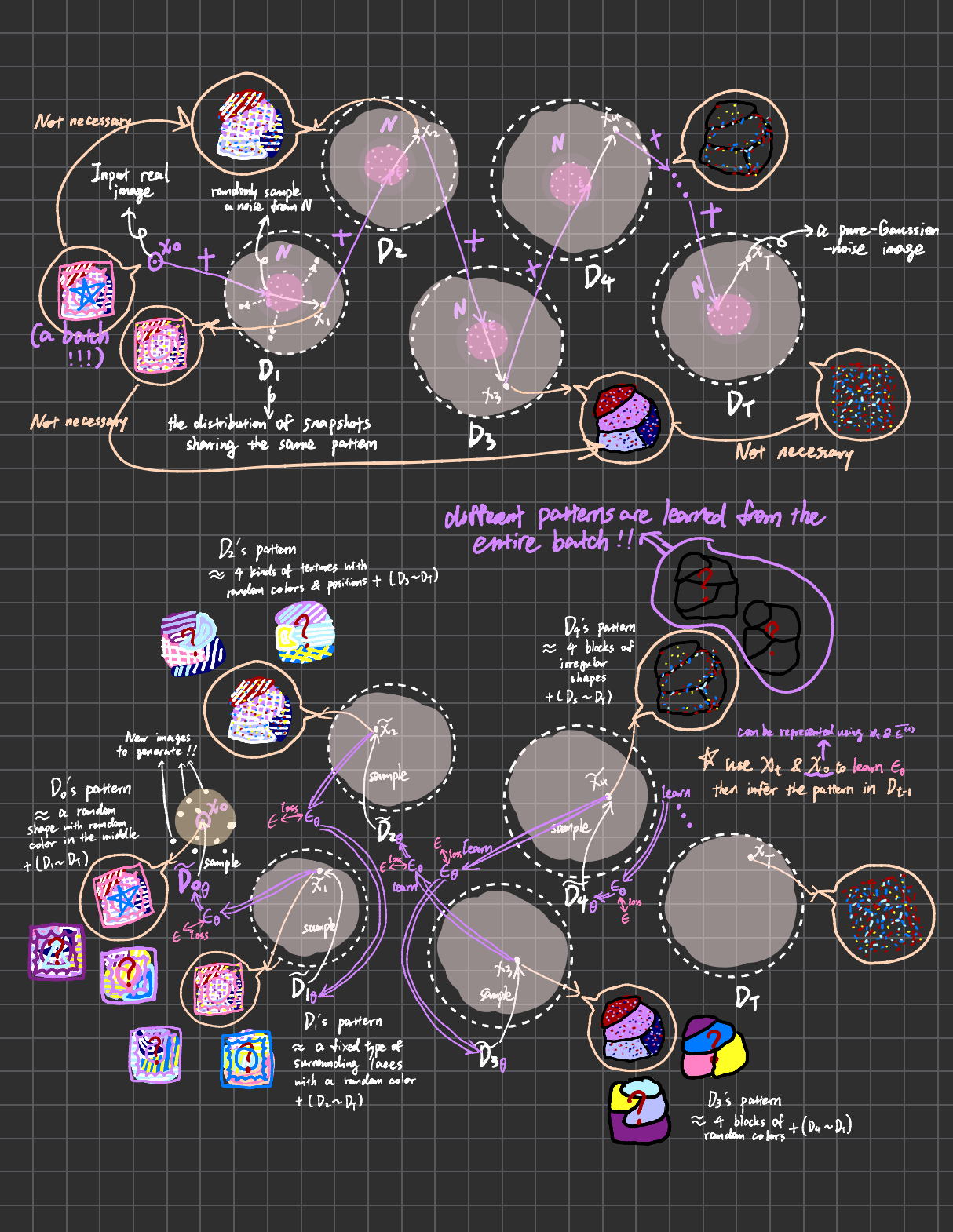
$$\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i \qquad \underbrace{\sqrt{\bar{\alpha}_t}}_{\mu}\cdot x_0 + \underbrace{\sqrt{1-\bar{\alpha}_t}}_{\sigma}\cdot \epsilon^{(t)} \quad (\epsilon^{(t)}: \text{Embed noise from } N(0,I) \text{ 't' times})$$

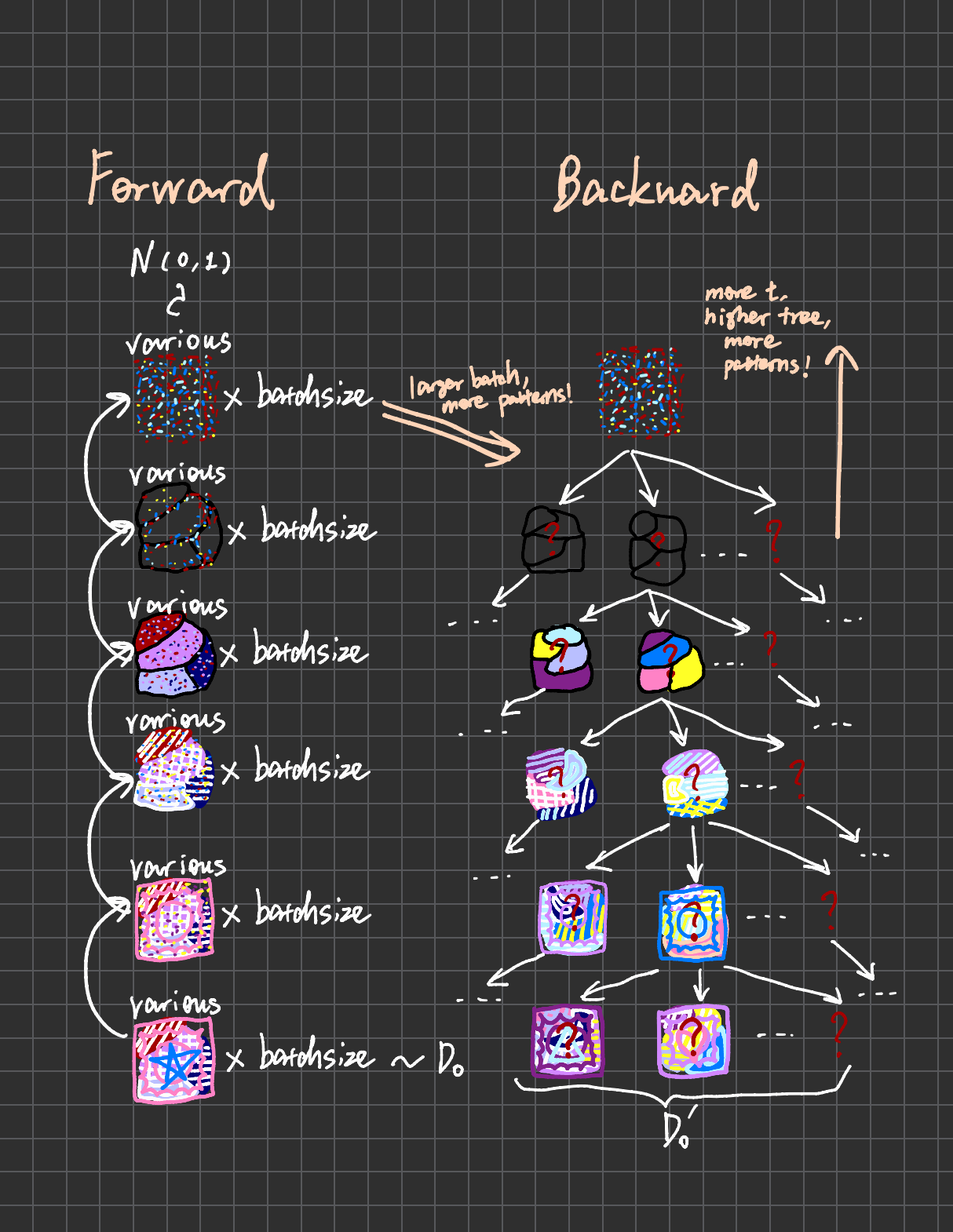
$$\Rightarrow q(x_t|x_0) = N(x_t; \sqrt{\bar{\alpha}_t}\cdot x_0, 1-\bar{\alpha}_t) \rightarrow \text{For Reverse Process ("Snapshots")}$$

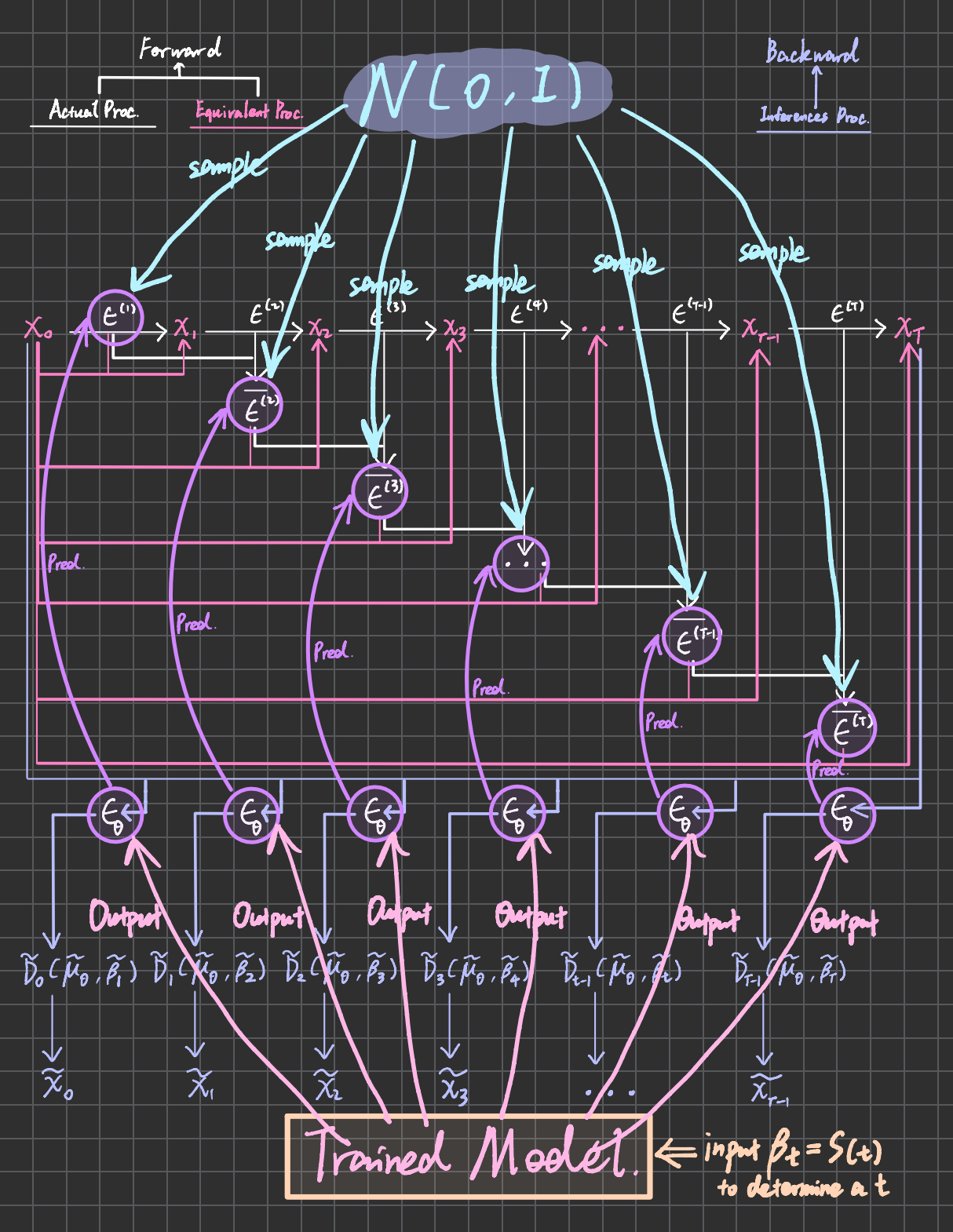
$$N(0,I) \sim \sqrt{(1-\beta_t)\beta_{t-1}}\cdot \epsilon^{(2)} + \sqrt{\beta_t}\cdot \epsilon^{(1)} \sim N(0, (1-\beta_t)\beta_{t-1} + \beta_t) = N(0, (1-\beta_t)\beta_{t-1} + \beta_t) = \sqrt{(1-\beta_t)\beta_{t-1}+\beta_t}\cdot N(0,I) \Leftrightarrow \sqrt{(1-\beta_t)\beta_{t-1}+\beta_t}\cdot \epsilon^{(2)} \sim N(0,I)$$

⚠ The snapshots are not for exactly anchoring the targets in Reversing !!
They're only the clues for the reverse model to land in their own distributions unknow to the model! → ① Record a $\bar{\epsilon}^{(t)}$; ② Visual features of x_t.




Not necessary
Input real image
(a batch !!!)
randomly sample a noise from N
the distribution of snapshots sharing the same pattern
Not necessary
a pure Gaussian noise image
Not necessary
different patterns are learned from the entire batch !!
D2's pattern
≈ 4 kinds of textures with random colors & positions + (D3~DT)
D4's pattern
≈ 4 blocks of irregular shapes + (D5~DT)
can be represented using xt & ε(t)
☆ use xt & x0 to learn εθ then infer the pattern in Dt-1
New images to generate !!
D0's pattern
≈ a random shape with random color in the middle + (D1~DT)
sample
learn
loss
D1's pattern
≈ a fixed type of surrounding trees with a random color + (D2~DT)
D3's pattern
≈ 4 blocks of random colors + (D4~DT)


Forward
Backward
N(0,1)
?
various
x batchsize
various
x batchsize
various
x batchsize
various
x batchsize
various
x batchsize
various
x batchsize ~ D0
larger batch, more patterns!
more t, higher tree, more patterns!
D0'


Forward
Actual Proc.
Equivalent Proc.
Backward
Inferences Proc.
N(0, I)
sample
sample
sample
sample
sample
sample
Preal.
Preal.
Preal.
Preal.
Preal.
Preal.
Output
Output
Output
Output
Output
Output
Trained Model.
input $\beta_t = S(t)$
to determine a t

Backward: denoise (generative)

Assume that: $p(x_{t-1}|x_t, x_{t+1}, \dots, x_T) = p(x_{t-1}|x_t)$

Designed → Markov Chain.

Assumed → $q(x_{t-1}|x_t) = N(x_{t-1}; \tilde{\mu}, \tilde{\beta}_t^2 I)$ because β_t is small enough.

> diffusion (limit of small step size β) the reversal of the diffusion process has the identical functional form as the forward process (Feller, 1949). Since $q(x^{(t)}|x^{(t-1)})$ is a Gaussian (binomial) distribution, and if β_t is small, then $q(x^{(t-1)}|x^{(t)})$ will also be a Gaussian (binomial) distribution. The longer the trajectory, the smaller the diffusion rate β can be made.

⇒ Need to be learned!! Known from forward proc.: $x_0, \bar{\epsilon}^{(t)}, x_t, \beta_t \Rightarrow q(x_t|x_{t-1}), q(x_t|x_0), x_T$

target a destination to denoise

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_{t-1}, x_t, x_0)}{q(x_t, x_0)} = \frac{q(x_t|x_{t-1}, x_0)\cdot q(x_{t-1}|x_0)}{q(x_t|x_0)} = \frac{q(x_t|x_{t-1})\cdot q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

eliminate the unknown $q(x_0)$ using Markov chain's assumption

$N(x_t; \sqrt{\bar{\alpha}_{t-1}}x_{t-1}, \beta_t I)$; $N(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0, 1-\bar{\alpha}_{t-1})$; keep the known x_0 & x_t (in $q(x_t|x_0)$)

$$= \frac{1}{\sqrt{2\pi}\cdot\beta_t}\cdot e^{\frac{(x_t-\sqrt{1-\beta_t}x_{t-1})^2}{-2\beta_t}} \cdot \frac{1}{\sqrt{2\pi}(1-\bar{\alpha}_{t-1})}\cdot e^{\frac{(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0)^2}{-2(1-\bar{\alpha}_{t-1})}} \Big/ \frac{1}{\sqrt{2\pi}(1-\bar{\alpha}_t)}\cdot e^{\frac{(x_t-\sqrt{\bar{\alpha}_t}x_0)^2}{-2(1-\bar{\alpha}_t)}} = \frac{\sqrt{1-\bar{\alpha}_{t-1}}}{\sqrt{2\pi}\cdot\beta_t\cdot(1-\bar{\alpha}_{t-1})}\cdot e^{-\frac{1}{2}\cdot\left[\frac{(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t-\sqrt{\bar{\alpha}_t}x_0)^2}{1-\bar{\alpha}_t} + \frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t}\right]}$$

$N(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot e^{\frac{(x-\mu)^2}{-2\sigma^2}}$

$$\frac{\sqrt{1-\bar{\alpha}_{t-1}}}{\sqrt{2\pi}\cdot\beta_t\cdot(1-\bar{\alpha}_{t-1})} = C_0 \qquad C_0\cdot e^{-\frac{1}{2}\cdot\left[\left(\frac{1}{1-\bar{\alpha}_{t-1}}+\frac{\alpha_t}{\beta_t}\right)\cdot x_{t-1}^2 - 2\left(\frac{x_0\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}+\frac{x_t\sqrt{\alpha_t}}{\beta_t}\right)\cdot x_{t-1} + C_1(x_t, x_0)\right]} \Leftrightarrow$$

Can be represented with $\beta_t, \bar{\alpha}_t, x_t$ but not enough!! ⇒ Model q with p_θ for learning

NEED TO KNOW THE DISTRIBUTION OF x_t is random!!!

$$\Longleftrightarrow \begin{cases} \frac{1}{1-\bar{\alpha}_{t-1}} + \frac{\alpha_t}{\beta_t} = \frac{1}{\tilde{\sigma}_t^2} \\ \frac{x_0\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} + \frac{x_t\sqrt{\alpha_t}}{\beta_t} = \frac{\tilde{\mu}_t}{\tilde{\sigma}_t^2} \end{cases} \Longleftrightarrow \begin{cases} \tilde{\sigma}_t^2 = \tilde{\beta}_t^2 = \frac{\beta_t(1-\bar{\alpha}_{t-1})}{\beta_t+\alpha_t(1-\bar{\alpha}_{t-1})} = \frac{\beta_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \\ \tilde{\mu}_t = \frac{\beta_t x_0\sqrt{\bar{\alpha}_{t-1}} + (1-\bar{\alpha}_{t-1})x_t\sqrt{\alpha_t}}{\beta_t+\alpha_t(1-\bar{\alpha}_{t-1})} = \frac{\beta_t x_0\sqrt{\bar{\alpha}_{t-1}} + (1-\bar{\alpha}_{t-1})x_t\sqrt{\alpha_t}}{1-\bar{\alpha}_t} \end{cases}$$

(Use the actual input)

$$\Rightarrow q(x_{t-1}|x_t, x_0) = N(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t^2 I)$$

☆ The Multi-person method from forward proc.

$x_t = \sqrt{\bar{\alpha}_t}\cdot x_0 + \sqrt{1-\bar{\alpha}_t}\cdot\epsilon^{(t)}$ ($x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\cdot\epsilon^{(t)}}{\sqrt{\bar{\alpha}_t}}$)

$$\tilde{\mu}_t = \frac{x_t}{\sqrt{\alpha_t}} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon^{(t)}$$

Random!

☆ better when x_0 is fixed

☆ better when x_0 is sampled

(☆ actually equivalent to $\epsilon^{(t)}$) ☆ learned

$$\begin{cases} \Sigma_\theta\ (:= \tilde{\beta}_t^2 \approx \beta_t \text{ in DDPM}) \\ \mu_\theta := \frac{x_t}{\sqrt{\alpha_t}} - \frac{\beta_t\cdot\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}} \end{cases} \Rightarrow \text{Loss}$$

$\epsilon^{(t)}$

& $\because \epsilon^{(t)} = \frac{x_t - \sqrt{\bar{\alpha}_t}\cdot x_0}{\sqrt{1-\bar{\alpha}_t}}$ is sampled & x_0 is given

$\therefore$ To be learnable, use $\epsilon_\theta(x_t, t)$ as an equivalence to the single random $\epsilon^{(t)}$

introduced by α_t ($\bar{\alpha}_t$)

☆ transform randomness into a structural rule (a function: $\epsilon_\theta = \epsilon_\theta(x_t, t; \theta)$)

$$\Rightarrow p_\theta(x_{t-1}|x_t) := N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta I) \xrightarrow{\text{learn}} q(x_{t-1}|x_t)$$

Markov Chain Assumption

$q(x_{t-1}|x_t, x_0)$

☆ Use the Multi-person Method instead of the original N to be differentiable.

And $\tilde{x}_{t-1} = \mu_\theta(x_t, t) + \sqrt{\Sigma_\theta}\cdot\epsilon^*$ ($\tilde{x}_{t-1}$: the next x_t)

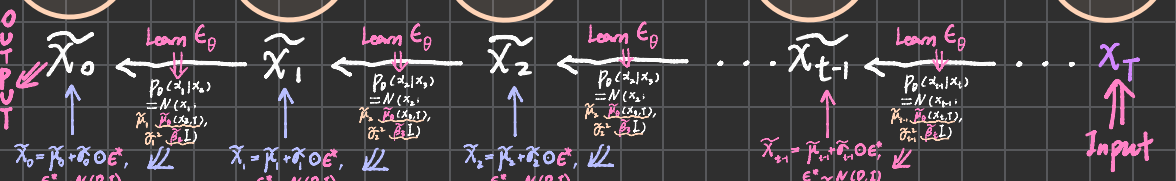
Still sampled from $N(0, I)$! (Added noise)

Learned! Contain ϵ_θ!!

Only used when Denoising! (Not in training)

☆ Ready for the Variational lower bound of $p_\theta(x_0)$!

OUTPUT



$$\Rightarrow p_\theta(x_{0:T}) = p_\theta(x_{1:T})\cdot p_\theta(x_0|x_{1:T}) = p_\theta(x_{2:T})\cdot p_\theta(x_1|x_{2:T})\cdot p_\theta(x_0|x_{1:T}) = \dots = p_\theta(x_{T-1:T})\cdot p_\theta(x_{T-2}|x_{T-1}, x_T)\cdot p_\theta(x_{T-3}|x_{T-2}, x_{T-1}, x_T)\dots p_\theta(x_0|x_{1:T})$$

$$= \underbrace{p(x_T)}_{q(x_T)}\cdot p(x_{T-1}|x_T)\cdot p_\theta(x_{T-2}|x_{T-1}, x_T)\cdots p_\theta(x_0|x_{1:T}) \quad \text{Markov Chain Assumption} \quad p(x_T)\cdot p_\theta(x_{T-1}|x_T)\cdot p_\theta(x_{T-2}|x_{T-1})\cdot\dots\cdot p_\theta(x_0|x_1)$$

$$= p(x_T)\cdot\Pi_{t=1}^T p_\theta(x_{t-1}|x_t) = p(x_T)\,\Pi_{t=1}^T N(x_{t-1}; \tilde{\mu}_\theta(x_t, t), \tilde{\beta}_\theta I) \Rightarrow \text{the Reverse Proc. (Denoise)}$$

$N(x_T; 0, I)$ (Pure noise)

Training.

Assumption: $q(x_t)$ is Gaussian

Learning: $p_\theta(x_t)$ learns the noise in q_t as a function

Ground-truth: p_t has $p(x_t) \approx 1$, which is the target of p_θ

Initial target: $\epsilon_\theta(x_t, t)$ equivalent to $\epsilon^{(t)} \sim N(t; 0, I)$

(⚠ No ways to calculate loss directly because equivalent is not exact "equal")

Practical target: $p_\theta(x_0) \to p(x_0)$ ← In actual Distribution of x_0

(⚠ Still difficult to optimize p_θ directly! Impossible to calculate prior prob.)

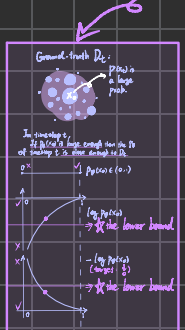
More Practical target:

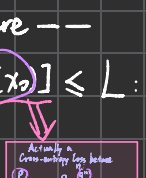
$\min L$ where ——

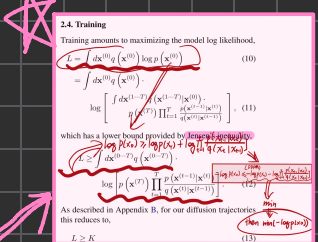
$$E[-\log p_\theta(x_0)] \le L := L_{VLB} = E_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

$$= E_q\left[-\log p_\theta(x_T) - \sum_{t\ge 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$

(means that using the simple Gaussian-q to calculate this lower bound by means of which to optimize the black-boxed ground-truth prior p.)







$$= E_q\left[-\log p_\theta(x_T) - \left(\sum_{t\ge 2} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} + \log \frac{p_\theta(x_1|x_0)}{q(x_1|x_0)}\right) - \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$= E_q\left[-\log p_\theta(x_T) - \sum_{t\ge 2} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} - \log \frac{q(x_1|x_0)}{q(x_T|x_0)} \cdot \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$= E_q\left[-\log p_\theta(x_T) + \sum_{t\ge 2} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} + \log \frac{q(x_T|x_0)}{p_\theta(x_0|x_1)}\right]$$

$$= E_q\left[\sum_{t\ge 1} \log \frac{q(x_{t-1}|x_t, x_0)}{p_\theta(x_{t-1}|x_t)} + \log \frac{q(x_T|x_0)}{p_\theta(x_T)} - \log p_\theta(x_0|x_1)\right]$$

(variance reduction using KL div.)

$$E_q\left[\underbrace{D_{KL}(q(x_T|x_0) \,\|\, p_\theta(x_T))}_{L_T \text{ (constant)}} + \sum_{t\ge 2} \underbrace{D_{KL}(q(x_{t-1}|x_t, x_0) \,\|\, p_\theta(x_{t-1}|x_t))}_{L_{t-1} \text{ (to be learnt)}} \underbrace{- \log p_\theta(x_0|x_1)}_{L_0 \text{ (modeled seperately with a discrete decoder derived from } N(x_0; \mu_\theta(x_1, 1), \Sigma_\theta(x_1, 1)))}\right]$$

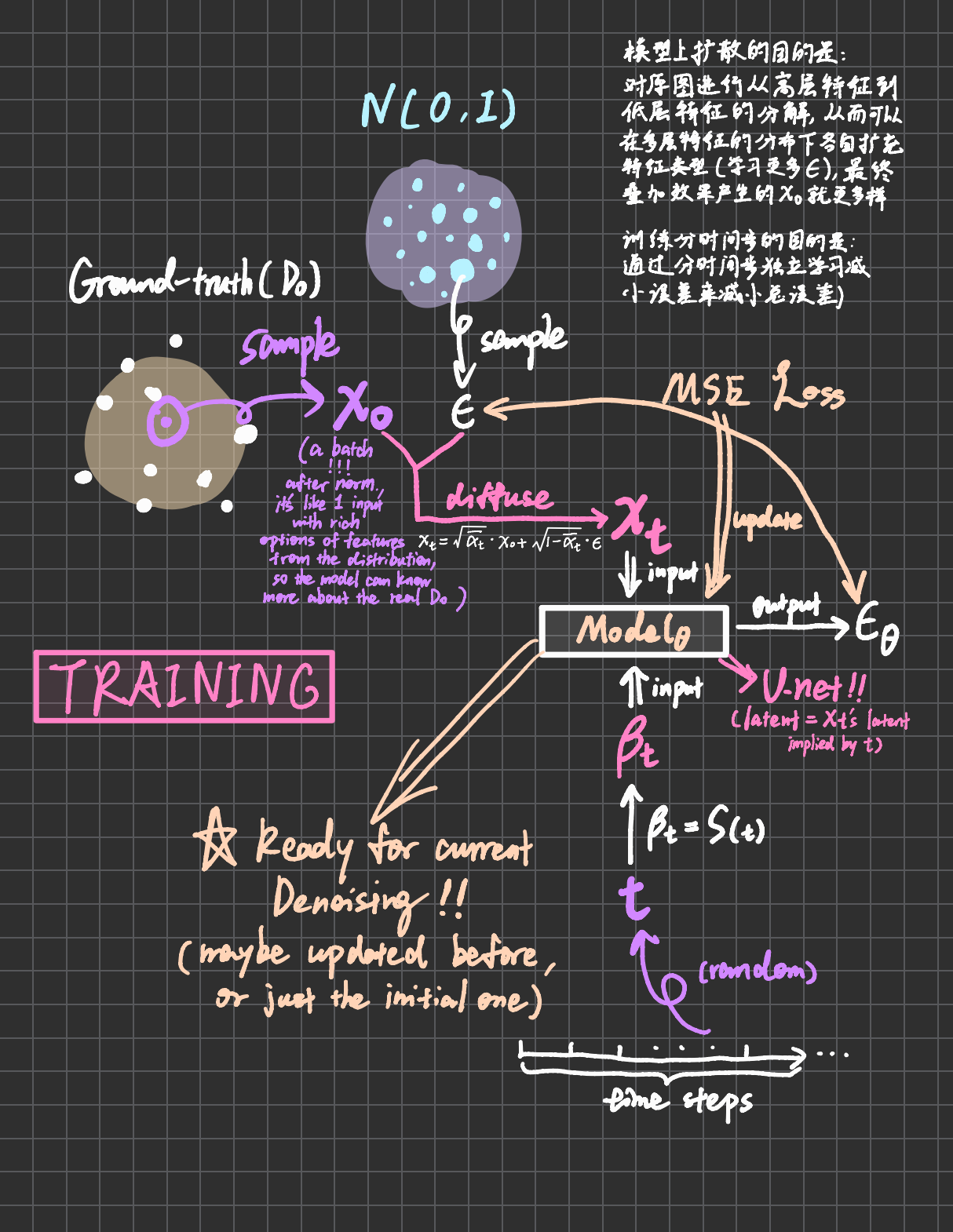
$$L_t \xrightarrow{D_{KL}(P\|Q) = \frac{1}{2}\left(\log \frac{|\Sigma_Q|}{|\Sigma_P|} - d + \mathrm{tr}(\Sigma_Q^{-1}\Sigma_P) + (\mu_Q - \mu_P)^T \Sigma_Q^{-1}(\mu_Q - \mu_P)\right)} E_{x_0,\epsilon}\left[\frac{1}{2\|\Sigma_\theta(x_t,t)\|_2^2} \left\| \tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t) \right\|^2\right]$$

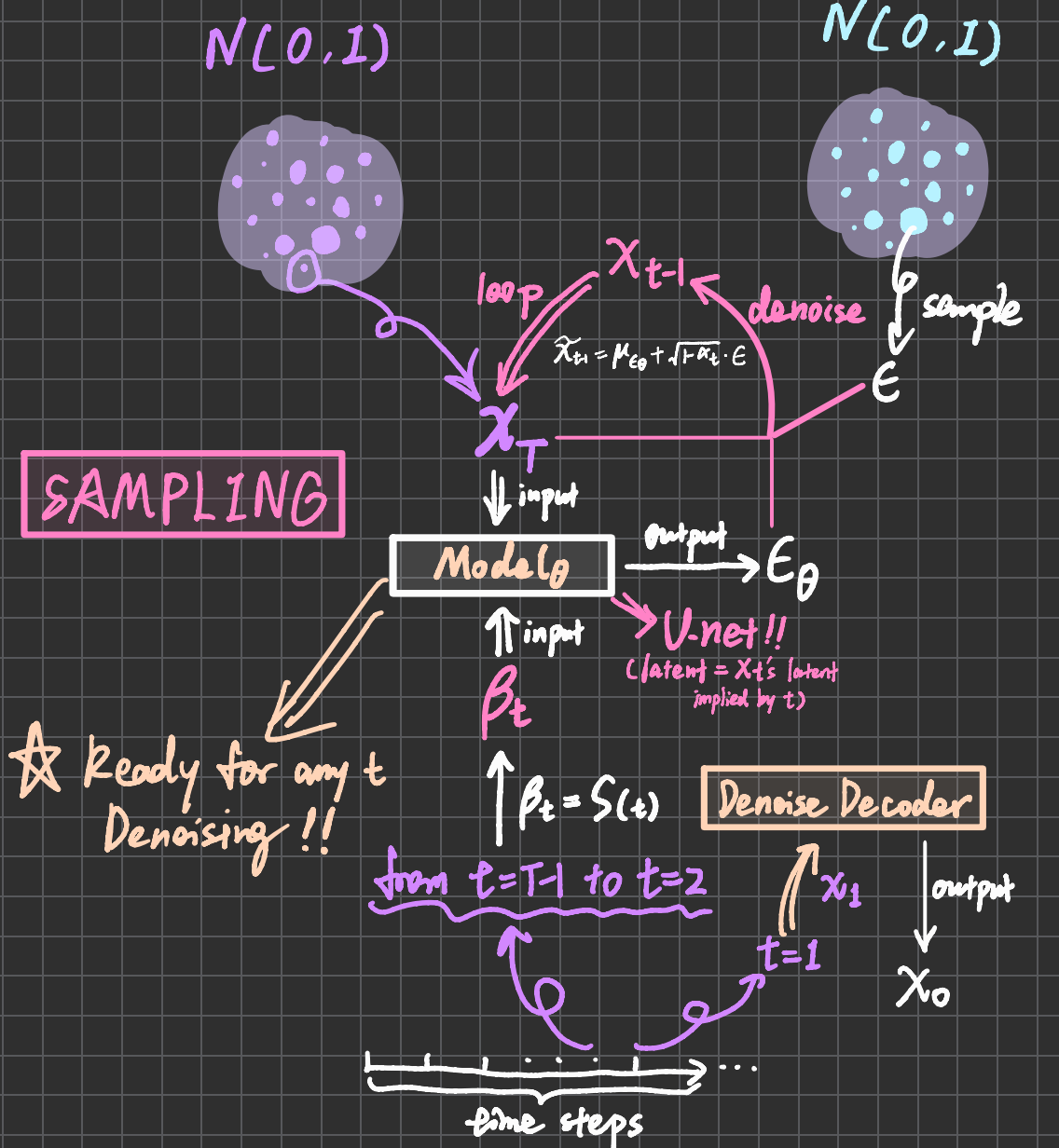
$$\xrightarrow{\substack{x_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t \cdot \epsilon^{(t)}}{\sqrt{1-\bar{\alpha}_t}}\right) \\ \mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t \cdot \epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}\right)}} E_{x_0,\epsilon}\left[\frac{1}{2\|\Sigma_\theta(x_t,t)\|_2^2} \left\| \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t \cdot \epsilon}{\sqrt{1-\bar{\alpha}_t}}\right) - \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t \cdot \epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}\right) \right\|^2\right]$$

Multi-process involved for x_t ($x_t = x_t(x_0, \epsilon)$)

$$\xrightarrow{}} E_{x_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta(x_t,t)\|_2^2} \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, t) \right\|^2\right]$$

(to speed up training)


模型上扩散的目的是：
对序图进行从高层特征到低层特征的分解，从而可以在多层特征的分布下各自扩充特征类型（学习更多ϵ），最终叠加效果产生的X₀就更多样
训练分时间步的目的是：
通过分时间步独立学习减小误差来减小总误差
N(0,1)
Ground-truth(D₀)
sample
sample
ϵ
MSE Loss
X₀
(a batch !!! after norm, its like 1 input with rich options of features from the distribution, so the model can know more about the real D₀)
diffuse
$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon$
Xt
update
input
Model_θ
output
ϵθ
TRAINING
input
U-net!!
(latent = Xt's latent implied by t)
βt
βt = S(t)
t
(random)
time steps
Ready for current Denoising !!
(maybe updated before, or just the initial one)


N(0,1)
N(0,1)
loop
x_{t-1}
denoise
$\tilde{x}_{t-1} = \mu_{\epsilon\theta} + \sqrt{1-\alpha_t} \cdot \epsilon$
sample
ϵ
x_T
SAMPLING
input
Model$_\theta$
output
ϵ_θ
input
U-net!!
(latent = x_t's latent implied by t)
β_t
Ready for any t Denoising!!
$\beta_t = S(t)$
Denoise Decoder
from t=T-1 to t=2
x_1
output
t=1
x_0
time steps