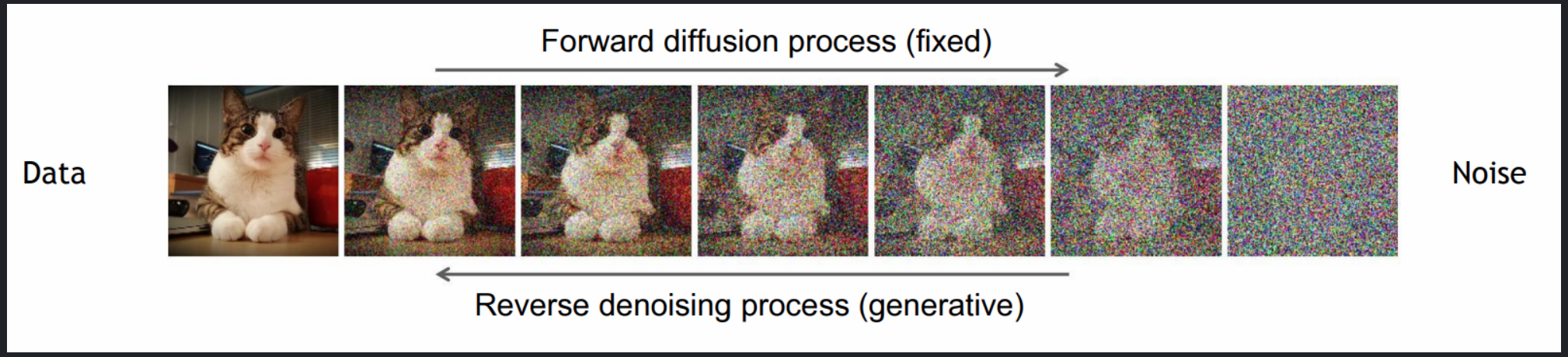
Diffusion Models

An Introduction

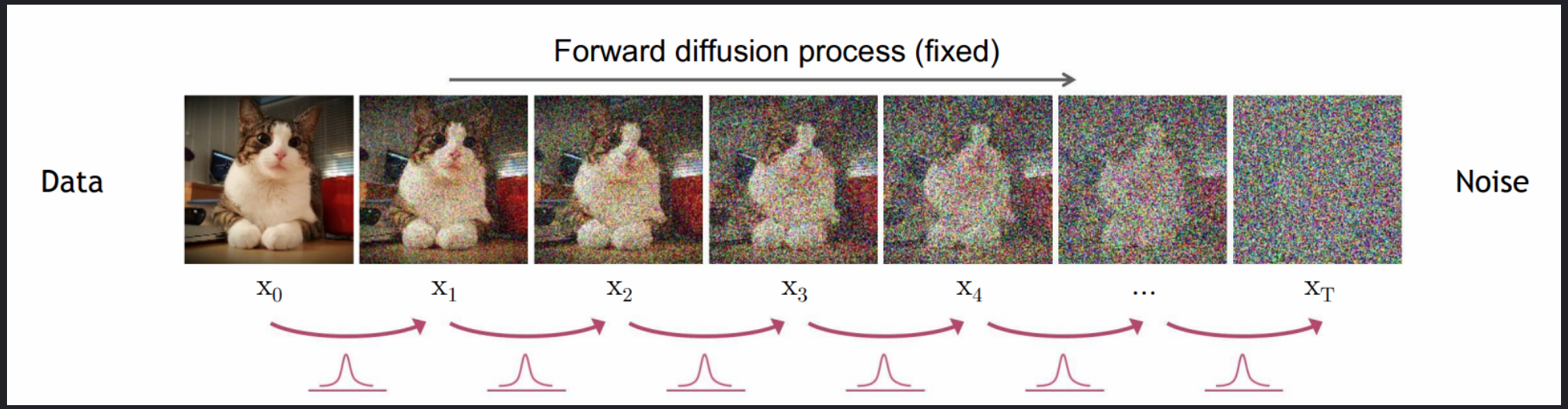
Learning to Generate by Denoising

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



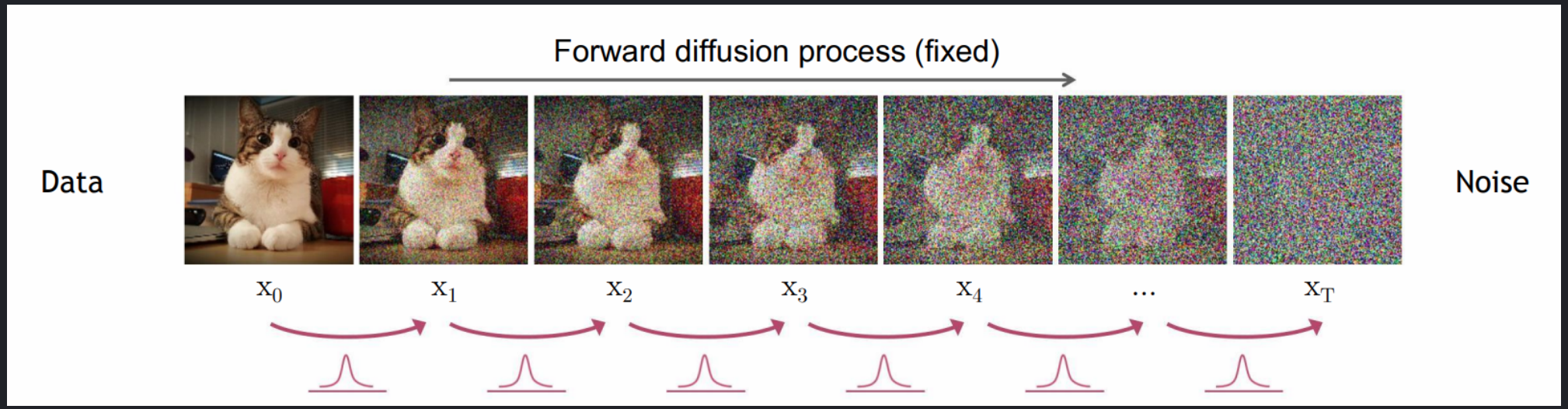
Forward Process

- **The definition of the forward process in T steps:**
  - Posterior: $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$,
  - where $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$



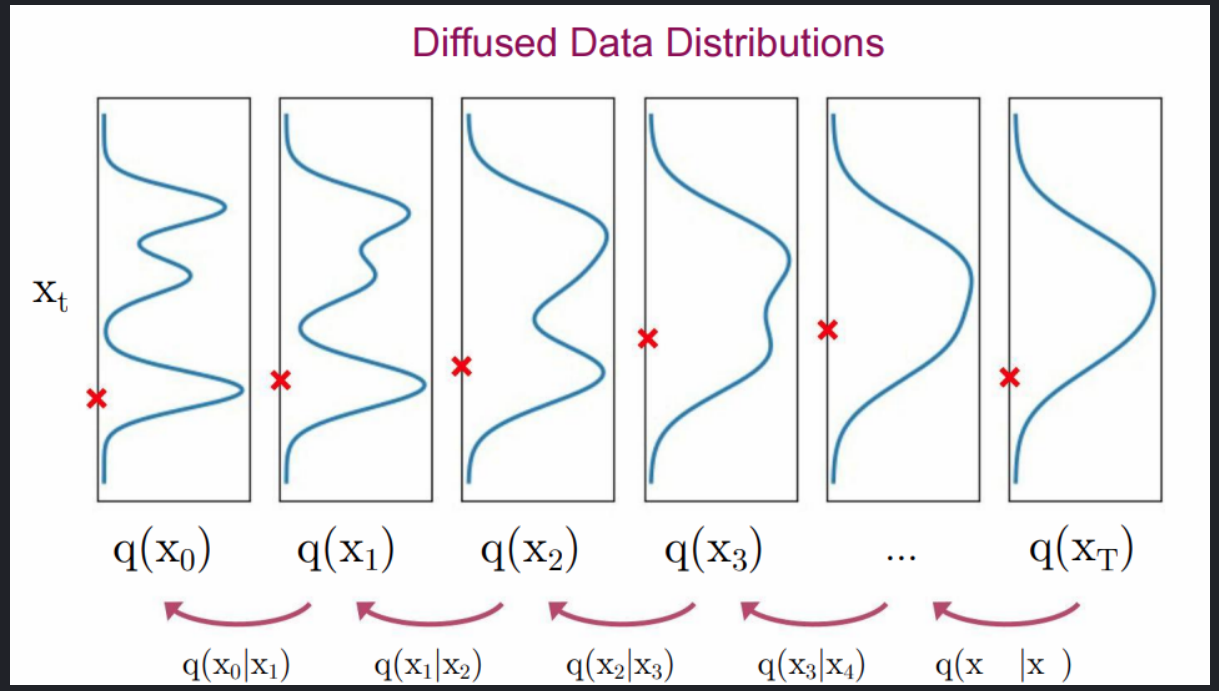
Characteristics of the forward process

- Markov Chain.
- Generally $\beta_t \in (0, 1)$ follows a fixed linearly increasing schedule.
- As $T \to \infty$, $q(x_t|x_0) \approx \mathcal{N}(0, I)$.
- If we set a large enough T we can set $\beta_t \ll 1$.



Backward Process

- We have to approximate $q(x_{t-1}|x_t)$.
- We can use model $q(x_{t-1}|x_t)$ as a Normal Distribution is β_t is small.

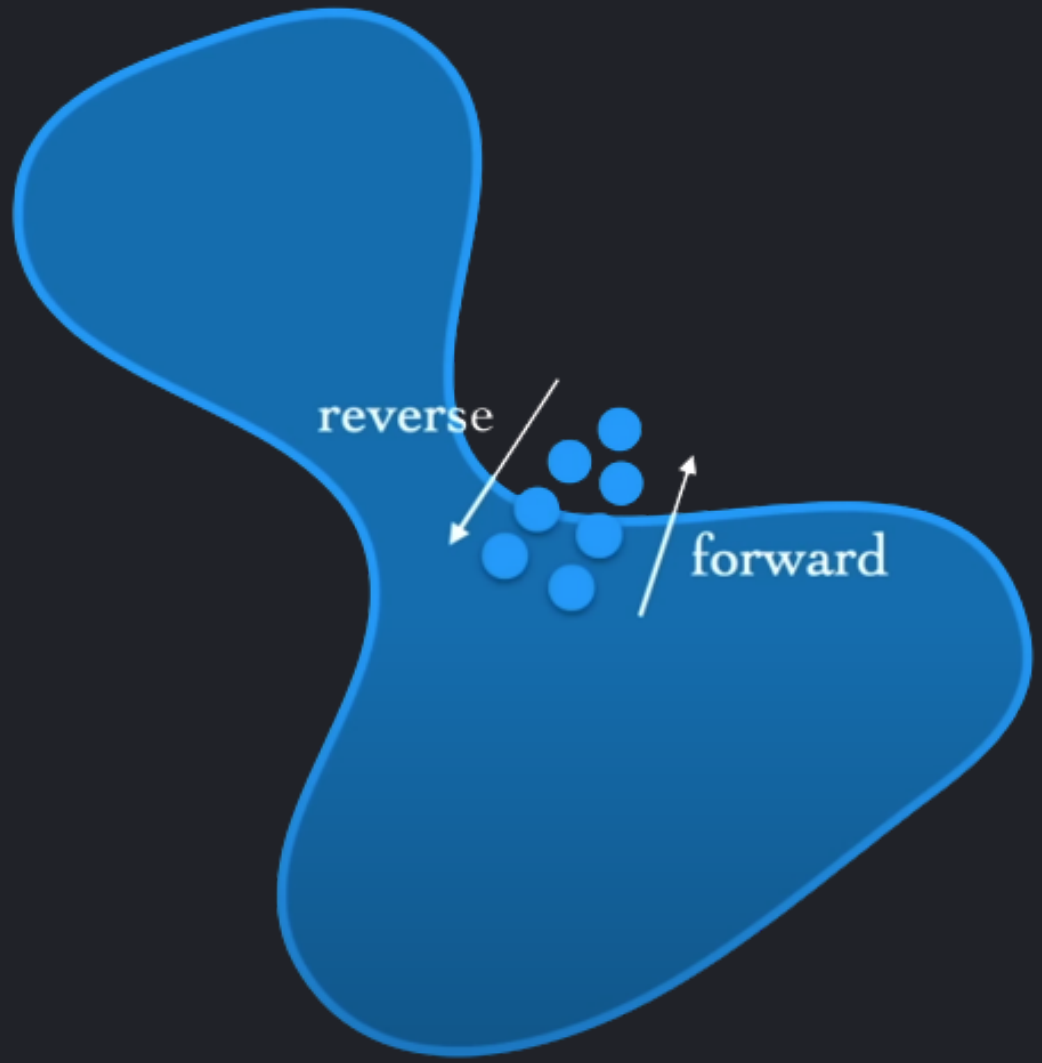


Definition of the Backward Process

- We will model the true reverce process $q(x_{t-1}|x_t)$ with $p_\theta(x_{t-1}|x_t)$.
- $p_\theta(x_T) = \mathcal{N}(x_T; 0, I)$.
- $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$

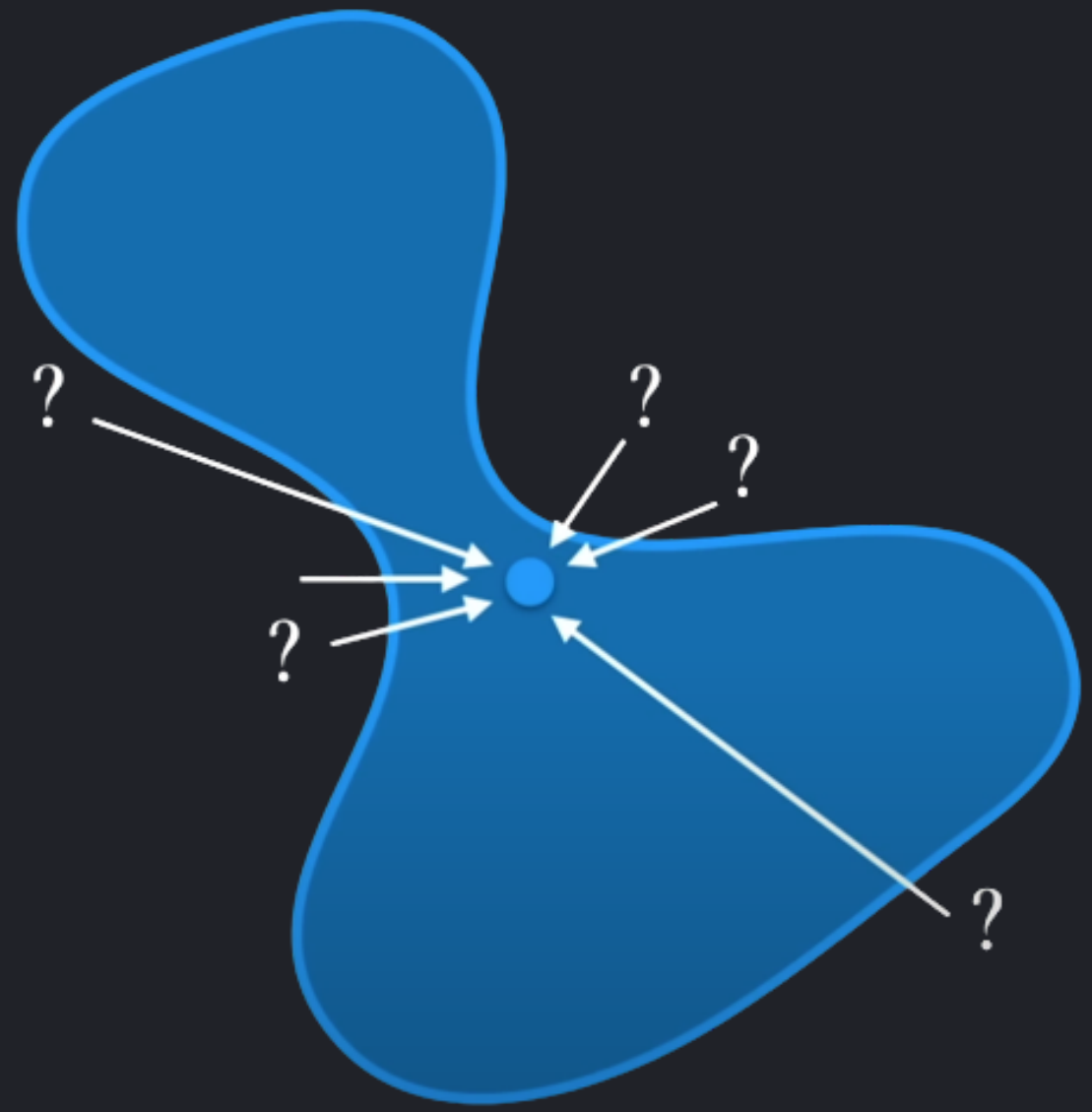
What objective will we optimize?

- Maximize $p_\theta(x_0)$?
- $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$

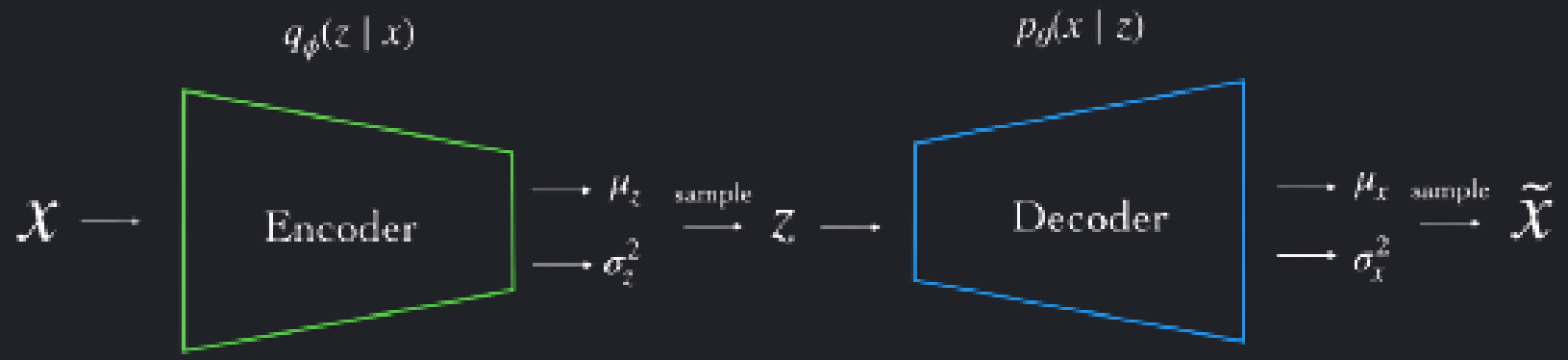


What objective will we optimize?

- Marginalizing oven all possile trajectories is intracable.
- $p_\theta(x_0) = \int p_\theta(x_{0:T}) \underline{dx_{1:T}}$



VAE



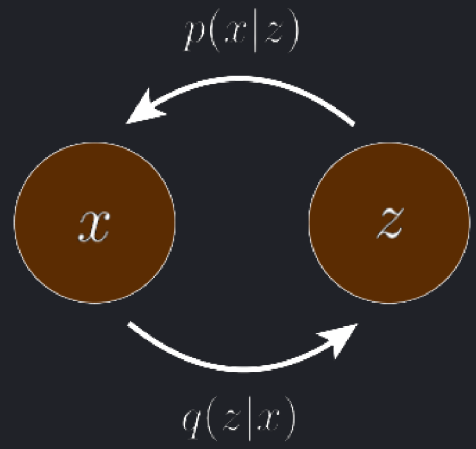
Diffusion model

$$x_0 \longrightarrow x_1 \longrightarrow \quad \longrightarrow x_T \longrightarrow x_{T-1} \longrightarrow \cdots \longrightarrow x_0$$

forward "encoding" reverse "decoding"

Mininize the Evidence Lower Bound (ELBO)

- The evidence is quantified as the log likelihood of the observed data.
- Maximizing the ELBO becomes a proxy objective with which to optimize a latent variable model.
- $\log p(x) \geq \mathbb{E}_{q_\phi(z|x)}[log(\frac{p(x,z)}{q_\phi(z|x)})]$



Derivation of the Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p(x) &= \int \log p(x) q_\phi(z|x) dz && (\int q(z|x)dz = 1) \\
&= \mathbb{E}_{q_\phi(z|x)}[\log p(x)] && (\text{By Definition of } \mathbb{E}) \\
&= \mathbb{E}_{q_\phi(z|x)}[\log \frac{p(x,z)}{p(z|x)}] && (\text{Chain rule}) \\
&= \mathbb{E}_{q_\phi(z|x)}[\log \frac{p(x,z)}{q_\phi(z|x)}] + \mathbb{E}_{q_\phi(z|x)}[\log \frac{q_\phi(z|x)}{p(z|x)}] && (\text{Multiple by } \frac{q_\phi(z|x)}{q_\phi(z|x)} \text{ and split}) \\
&= \mathbb{E}_{q_\phi(z|x)}[\log \frac{p(x,z)}{q_\phi(z|x)}] + D_{KL}(q_\phi(z|x)||p(z|x)) && (\text{By definition of } D_{KL}) \\
&\geq \mathbb{E}_{q_\phi(z|x)}[\log \frac{p(x,z)}{q_\phi(z|x)}] && (D_{KL} > 0)
\end{aligned}
$$

Diffusion Models as Hierarchical VAEs

- The latent dimension is exactly equal to the data dimension.
- The the latent encoders q not learned; it is pre-defined as a linear Gaussian model.
- The Gaussian parameters of the latent encoders vary over time in such as the latent at final timestep T is $\mathcal{N}(0, I)$.

An ELBO for the Diffusion Model

$$\log p(x) \geq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[log \frac{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1}, x_0)}\right]$$

(Due to Markovian Property $q(x_t|x_{t-1}) = q(x_t|x_{t-1}, x_0)$)

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_T)p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1}, x_0)}\right]$$

(1st term out of product and split.)

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_T)p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{\frac{q(x_{t-1}|x_T, x_0)q(x_t|x_0)}{q(x_{t-1}|x_0)}}\right]$$

(Bayes rule)

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_T)p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log(\frac{q(x_1|x_0)}{q(x_T|x_0)}) + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1}, x_0)}\right]$$

(Only the first and last term survive.)

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

(Get log inside product)

An ELBO for the Diffusion Model

$$
\begin{aligned}
\log p(x) \geq & \mathbb{E}_{q(x_1|x_0)}\left[\log p_\theta(x_0|x_1)\right] && (L_0: \text{ Reconstruction term}) \\
& - D_{KL}(q(x_T|x_0)||p(x_T)) && (L_T: \text{Prior matchng term}) \\
& - \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))\right] && (L_{t-1}: \text{Denoising matching term})
\end{aligned}
$$

- L_0: Can be approximated and optimized using a Monte Carlo estimate.
- L_T: Equal to zero under our assumptions since $q(x_T|x_0) \approx p(x_T) = \mathcal{N}(0, I)$
- L_{t-1}: Denoising transition step $p_\theta(x_{t-1}|x_t)$ is learned as an approximation to tractable, ground-truth denoising transition step $q(x_{t-1}|x_t, x_0)$.

What objective will we optimize?

- In this derivation the bulk of the cost lies in L_{t-1}.
- If we find a closed form for $q(x_{t-1}|x_t, x_0)$ we can compute the KL Divergance and maximize the ELBO.
- By Bayes rule we have $q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}$.
- We already know that $q(x_t|x_{t-1}, x_0) = q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{a_t}x_t, (1 - a_t)I)$
- What about $q(x_t|x_0)$ and $q(x_{t-1}|x_0)$?

Reparameterization trick

Under the reparameterization trick, samples $x_t \sim q(x_t|x_{t-1})$ can be rewritten as:

$$x_t = \sqrt{a_t}x_{t-1} + \sqrt{1-a_t}e \qquad \text{with } e \sim \mathcal{N}(0,1)$$

Thus, $q(x_t|x_0)$ can be recursively derived through repeated applications of the reparameterization trick. Assuming acess to $2T$ random variables $\{e_t^*, e_t\}_{t=0}^T \sim \mathcal{N}(0,1)$.

$$\begin{aligned}
x_t &= \sqrt{a_t}x_{t-1} + \sqrt{1-a_t}e_{t-1}^* \\
&= \sqrt{a_t}\left(\sqrt{a_{t-1}}x_{t-2} + \sqrt{1-a_{t-1}}e_{t-2}^*\right) + \sqrt{1-a_t}e_{t-1}^* \\
&= \sqrt{a_t a_{t-1}}x_{t-2} + \sqrt{a_t - a_t a_{t-1}}e_{t-2}^* + \sqrt{1-a_t}e_{t-1}^* \\
&= \sqrt{a_t a_{t-1}}x_{t-2} + \sqrt{1-a_t a_{t-1}}e_{t-2} \qquad \text{(Sum of Gaussians)} \\
&= \quad \dots \\
&= \sqrt{\bar{a}_t}x_0 + \sqrt{1-\bar{a}_t}e_0
\end{aligned}$$

Now we can calculate the form of $q(x_{t-1}|x_t, x_0)$ ...

$$
\begin{aligned}
q(x_{t-1}|x_t, x_0) &= \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&= \frac{\mathcal{N}(x_t; \sqrt{a_t}x_t, (1-a_t)I)\ \ \mathcal{N}(x_{t-1}; \sqrt{\bar{a}_{t-1}}x_0, (1-\bar{a}_{t-1})I)}{\mathcal{N}(x_t; \sqrt{\bar{a}_t}x_0, (1-\bar{a}_t)I)} \\
&= \ \ldots \\
&= \mathcal{N}(x_{t-1}; \underbrace{\frac{\sqrt{a_t}(1-\bar{a}_{t-1})x_t + \sqrt{\bar{a}_{t-1}}(1-a_t)x_0}{1-\bar{a}_t}}_{\mu_q(x_t, x_0)}, \underbrace{\frac{(1-a_t)(1-\bar{a}_{t-1})}{(1-\bar{a}_t)}}_{\sum_q(t)} I)
\end{aligned}
$$

. . . and maximize the ELBO by minimizing the D_{KL}

$$
\begin{aligned}
& \text{argmin}_\theta \; D_{KL}(q(x_{t-1}|x_t, x_0) \;||\; p_\theta(x_{t-1}|x_t)) \\
= \; & \text{argmin}_\theta \; D_{KL}(\mathcal{N}(x_{t-1}; \mu_q(t), \Sigma_q(t)) \;||\; \mathcal{N}(x_{t-1}; \mu_\theta(t), \Sigma_q(t))) && \text{(set denoising transition variance to be } \Sigma_q(t)) \\
= \; & \; \ldots && \text{(KL Divergence Gaussians)} \\
= \; & \text{argmin}_\theta \; \frac{1}{2\sigma_q^2(t)} \left[||\mu_\theta - \mu_q||_2^2 \right]
\end{aligned}
$$

We want to optimize $\mu_\theta(x_t, t)$ to matches $\mu_q(x_t, x_0)$.

. . . and maximize the ELBO by minimizing the D_{KL}

We can match μ_θ and μ_q as close as possible:

$$\mu_q(x_t, x_0) = \frac{\sqrt{a_t}(1-\bar{a}_{t-1})x_t + \sqrt{\bar{a}_{t-1}}(1-a_t)x_0}{1-\bar{a}_t},$$

$$\mu_\theta(x_t, t) = \frac{\sqrt{a_t}(1-\bar{a}_{t-1})x_t + \sqrt{\bar{a}_{t-1}}(1-a_t)\mathbf{x}_\theta(\mathbf{x}_t, \mathbf{t})}{1-\bar{a}_t}$$

$x_\theta(x_t, t)$ is parameterized by a neural network that seeks to predict x_0 from noisy image x_t and time index t. So finaly we can write:

$$\text{argmin}_\theta = \frac{\bar{a}_{t-1}(1-a_t)^2}{2\sigma_q^2(t)(1-\bar{a}_t)^2}\left[||x_\theta(x_t, t) - x_0||_2^2\right]$$

Simplified Loss

Since x_t is available as input to the model, we can choose the parameterization:

$$x_0 = \frac{x_t - \sqrt{1-\bar{a}_t}\epsilon_0}{\sqrt{\bar{a}_t}}$$

μ_θ and μ_q become:

$$\mu_q(x_t, x_0) = \frac{1}{\sqrt{a_t}} x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}\sqrt{a_t}} \epsilon_0$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{a_t}} x_t - \frac{1-a_t}{\sqrt{1-\bar{a}_t}\sqrt{a_t}} \epsilon_\theta(x_t, t)$$

Reformulate the loss to:

$$\text{argmin}_\theta = \underbrace{\frac{(1-a_t)^2}{2\sigma_q^2(t)(1-\bar{a}_t)a_t}}_{\lambda_t} \left[||e_0 - e_\theta(x_t, t)||_2^2 \right]$$

- However, λ_t is often very large for small t's.
- Discard λ_t and minimize a weighted version of the ELBO.

Simplified Loss

$$\mathcal{L}_{Simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[||\epsilon - \epsilon_\theta(x_t)||_2^2\right]$$

DDPM: Training and Sampling

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** $\mathbf{x}_0$

DDPM: Implementation details

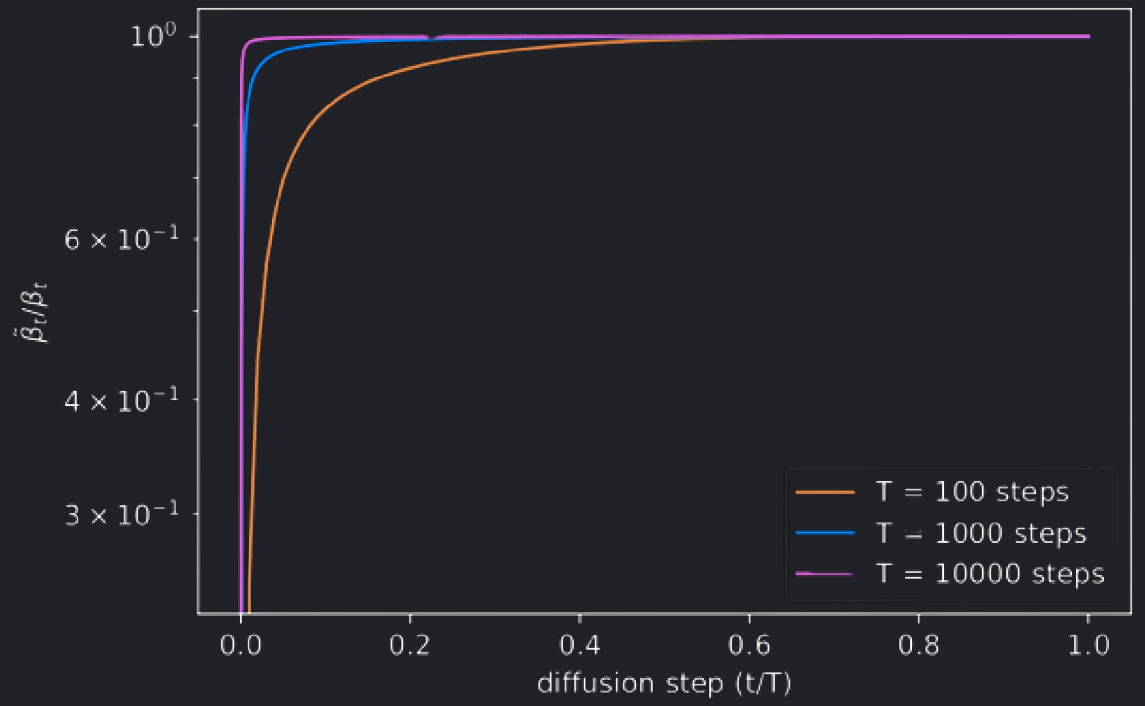
- A U-Net is used to estimate ϵ_θ
- Time information added to U-Net with positional embeddings
- Images scaled linearly to $[-1, 1]$
- Linear schudle from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$, with $T = 1000$
- These hyperparameters enusre that $a_T \to 0$ and $q(x_T|x_0) \approx \mathcal{N}(0, 1)$

DDPM: Predicting μ vs predicting ϵ

Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal Σ	7.28 ± 0.10	23.69
L, fixed isotropic Σ	8.06 ± 0.09	13.22
$\|\tilde{\mu} - \tilde{\mu}_\theta\|^2$	–	–
ϵ prediction (ours)		
L, learned diagonal Σ	–	–
L, fixed isotropic Σ	7.67 ± 0.13	13.51
$\|\tilde{\epsilon} - \epsilon_\theta\|^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**

Learning $\Sigma_\theta(x_t, t)$

- In DDPM $\Sigma_\theta(x_t, t) = \Sigma(t) = \sigma_t^2 I$
  - with $\sigma_t^2 = \beta_t$ or $\sigma_t^2 = \bar{\beta}_t := \frac{1-\bar{a}_{t-1}}{1-\bar{a}_t}\beta_t$
  - β_t and $\bar{\beta}_t$ yield similar results.
- β_t and $\bar{\beta}_t$ almost equal except near $t = 0$
- Learn to intertpolate β_t and $\bar{\beta}_t$
  - $\sigma_t^2 = \exp(v \log \beta_t + (1 - v) \log \bar{\beta}_t)$



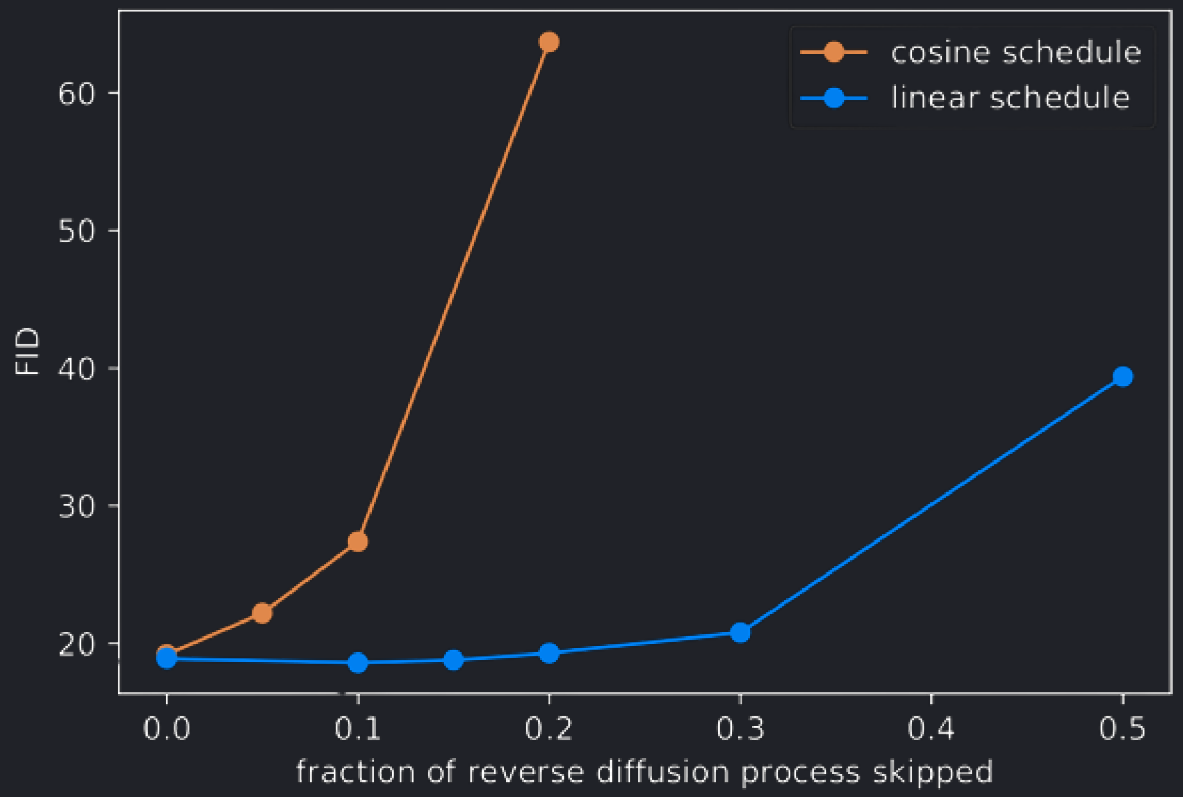
Alex Nichol et al., 2021, Improved Denoising Diffusion Probabilistic Models

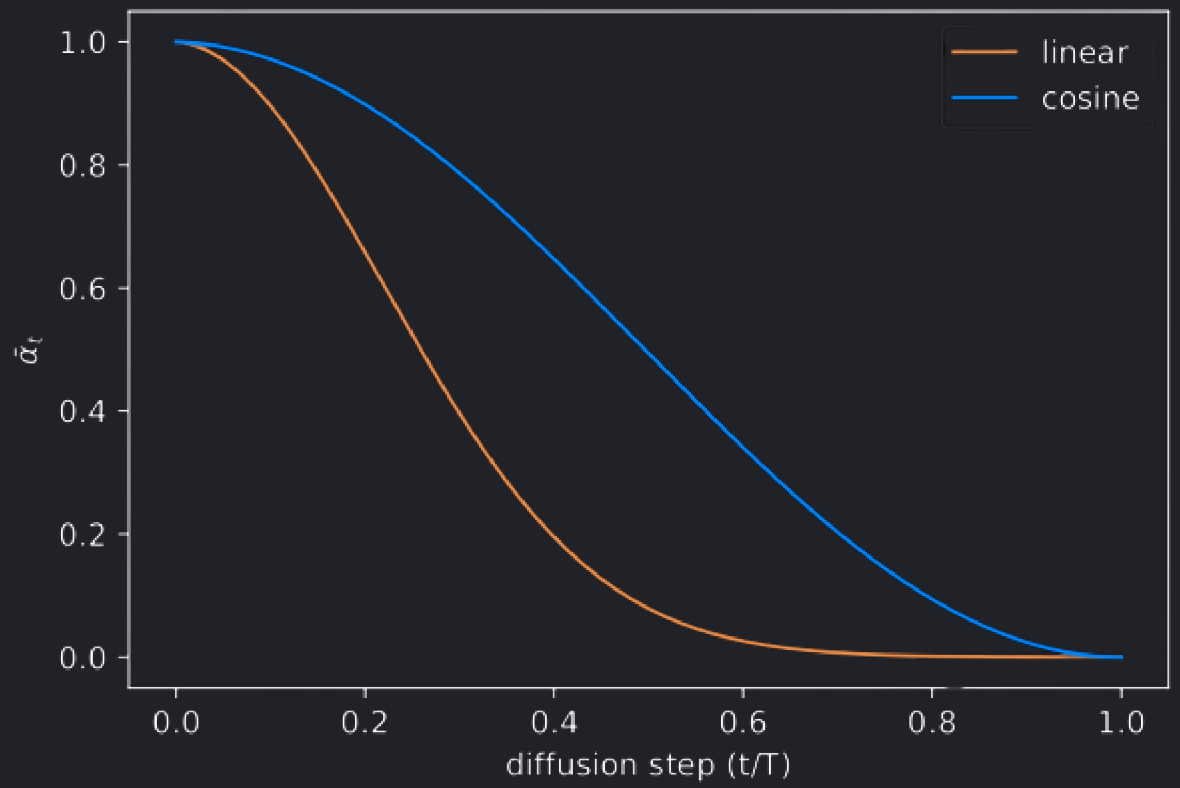
Noise Schedule

- Problems with linear schedule.
  - End of the forward noising process is too noisy.
  - Doesn’t contribute very much to sample quality
- Cosine schedule:
  - $\bar{a}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos^2(\frac{t/T+s}{1+s} \cdot \frac{\pi}{2})$

Alex Nichol et al., 2021, Improved Denoising Diffusion Probabilistic Models

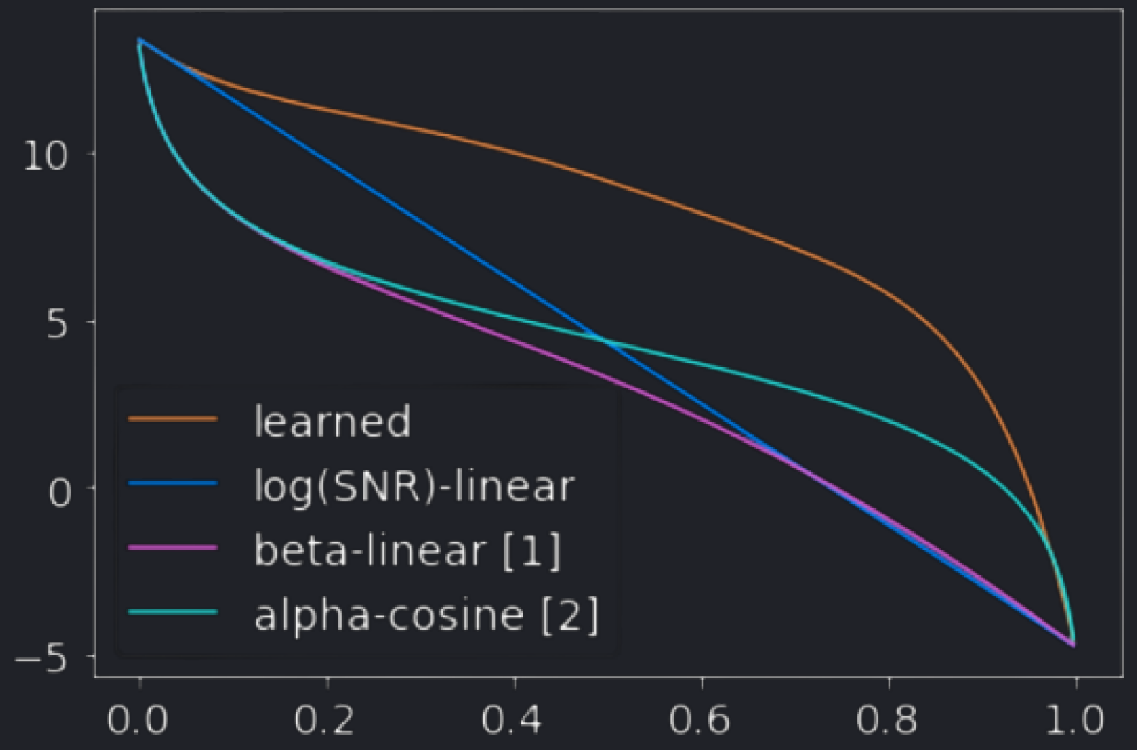
Noise Schedule: Cosine vs Linear





Learning the Noise Schedule

- Recall $q(x_t|x_0) = \mathcal{N}(x_t;\ \sqrt{\bar{a}_t}x_0, (1-\bar{a})I)$
- Define $\text{SNR} = \frac{\mu^2}{\sigma^2} = \frac{\bar{a}_t}{1-\bar{a}_t}$
- Model SNR with a NN:
  - $\text{SNR}(t) = \exp(-\omega_\phi(t))$
  - $\bar{a}_t = \text{sigmoid}(-\omega_\phi(t))$



(a) log SNR vs time t

D. Kingma et al., 2021, Variational diffusion models.