Denoising Diffusion Probabilistic Models (DDPMs)

Presented by
MEDIOCRE_GUY

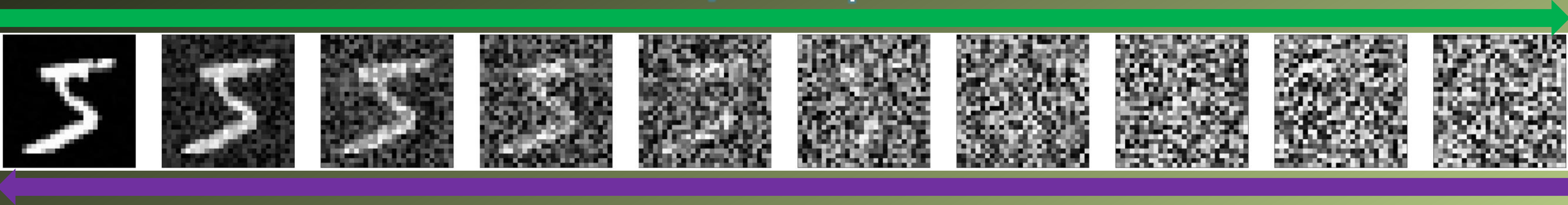
April 4, 2024

Introduction (1/2)

In simple terms, two processes happen in denoising diffusion probabilistic models (DDPMs):

- The data structure is destroyed by gradually adding Gaussian noise over a finite number of time steps to end up with pure noise (forward/diffusion process)
- A neural network is trained to gradually denoise the data starting from pure noise and predict a distribution that looks like the original distribution (reverse/denoising process)

Forward/diffusion process



Reverse/denoising process

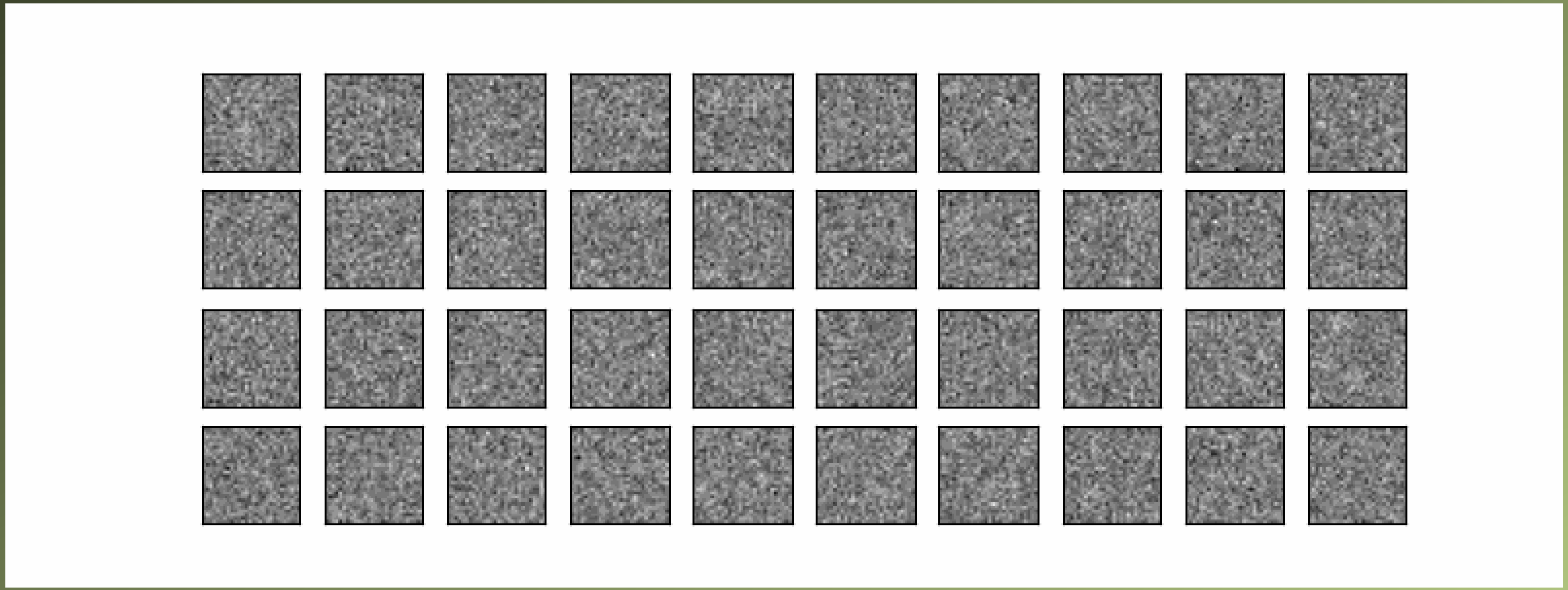
x_0 Original data | x_1 | ..... | x_{T-1} | x_T Pure noise

$p_0(x_0)$ → Becomes noise (in the forward process)

$p_T(x_T)$ → Turns into data (in the reverse process)

Introduction (2/2)

The objective of using diffusion models is to successfully generate actual images from pure noise only if they are trained well



Source: https://github.com/TeaPearce/Conditional_Diffusion_MNIST

Forward (Diffusion) Process (1/2)

$q(x_0)$ = Original data distribution

$x_0 \sim q(x_0)$

Taking a sample from the original data distribution

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

Forward process

Mean (μ_t): $\sqrt{1-\beta_t}x_{t-1}$

Variance (σ_t^2): β_t

x_{t-1} = Less noisy image

x_t = More noisy image

β_t = *Variance scheduler* (linear, cosine, sigmoid, quadratic etc.)

Forward process $q(x_t|x_{t-1})$ adds Gaussian noise $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ according to a known variance schedule ($0 < \beta_t < 1$)

β_t are constants that increase over T time steps

Original DDPM paper used *linear* scheduler ($\beta_1 = 0.0001$ to $\beta_T = 0.02$ for $T = 1000$)

The source image (x_0) eventually turns into pure noise (x_T) *through the forward process*

Source: https://arxiv.org/pdf/2006.11239.pdf

Forward (Diffusion) Process (2/2)

$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$ → A single step of the forward process

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$ → Equation for the *full forward process*

Reparameterization trick → $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma \odot \varepsilon$

$\odot$ → Element-wise product

$$\begin{aligned} x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\varepsilon \\ &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\varepsilon \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\varepsilon \\ &\ldots\ldots \\ &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon \end{aligned}$$

$\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\alpha_t = 1 - \beta_t$

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

For example, $\bar{\alpha}_3 = \alpha_1 . \alpha_2 . \alpha_3$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon$$

$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

This allows to sample x_t at any time step t conditioned on x_0

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (1/15)

$p(x_{t-1}|x_t)$

Reverse process

$p_\theta(x_{t-1}|x_t)$

The *reverse process* can not executed like the forward process because it requires knowing the distribution of all the images *in order*, which is impossible

Hence, a neural network is used to approximate the *denoising* part

$\theta \rightarrow$ all the parameters of the neural network

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\big(x_{t-1};\ \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\big)$$

Both the mean (μ_θ) and the variance (Σ_θ) are conditioned on the *noise level* (time step) t

In the original DDPM paper, the authors kept the variance (Σ_θ) fixed and used one neural network to learn only the mean (μ_θ)

$$\Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}$$

$$\sigma_t^2 = \beta_t$$

$$\sigma_t^2 = \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$$

DDPM authors claimed that both choices yielded *similar results*

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (2/15)

To learn the mean (μ_θ), the DDPM authors decided to minimize the negative log-likelihood (NLL) regarding the source image as their *objective function*

$$-\log\left(p_\theta(x_0)\right)$$

Not computable

Have to keep track of $T-1$ other random variables $(\mathrm{x}_T, x_{T-1}, x_{T-2}, \ldots, x_3, x_2, x_1)$

So, the DDPM authors chose to compute the *variational lower bound (VLB)* instead

$$-\log\left(p_\theta(x_0)\right) \leq -\log\left(p_\theta(x_0)\right) + \boxed{D_{KL}\left(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0)\right)}$$

Needs to be minimized

KL divergence:

$$D_{KL}(p||q) = \sum_x p(x)\log\left(\frac{p(x)}{q(x)}\right)$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (3/15)

$$D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0)) \Rightarrow \log\left(\frac{q(x_{1:T}|x_0)}{\boxed{p_\theta(x_{1:T}|x_0)}}\right)$$

$$p_\theta(x_{1:T}|x_0) = \underbrace{\frac{p_\theta(x_0|x_{1:T})p_\theta(x_{1:T})}{p_\theta(x_0)}}_{\text{Bayes' rule}} = \frac{\overbrace{p_\theta(x_0, x_{1:T})}^{\text{Joint probability}}}{p_\theta(x_0)} = \frac{p(x_{0:T})}{p_\theta(x_0)}$$

$$\log\left(\frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)}\right) = \log\left(\frac{q(x_{1:T}|x_0)}{\frac{p(x_{0:T})}{p_\theta(x_0)}}\right) = \underbrace{\log\left(\frac{q(x_{1:T}|x_0)}{p(x_{0:T})}\right) + \log(p_\theta(x_0))}_{\text{Log product rule}}$$

$$-\log(p_\theta(x_0)) \le -\log(p_\theta(x_0)) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0))$$ **now becomes**

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (4/15)

$$-\log\left(p_\theta(x_0)\right) \leq \boxed{-\log\left(p_\theta(x_0)\right)} + \log\left(\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right) \boxed{+\log\left(p_\theta(x_0)\right)}$$

$$-\log\left(p_\theta(x_0)\right) \leq \log\left(\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right)$$ *Variational Lower Bound (VLB)*

$q(x_{1:T}|x_0)$ is the *forward process*

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$\log\left(\frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right) = \log\left(\frac{\prod_{t=1}^{T} q(x_t|x_{t-1})}{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}\right) = -\log\left(p(x_T)\right) + \sum_{t=1}^{T}\log\left(\frac{q(x_t|x_{t-1})}{p_\theta(x_{t-1}|x_t)}\right)$$

$$= -\log\left(p(x_T)\right) + \sum_{t=2}^{T}\log\left(\frac{\boxed{q(x_t|x_{t-1})}}{p_\theta(x_{t-1}|x_t)}\right) + \log\left(\frac{q(x_1|x_0)}{p_\theta(x_0|x_1)}\right)$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (5/15)

$$q(x_t|x_{t-1}) = \frac{q(x_{t-1}|x_t)q(x_t)}{q(x_{t-1})}$$ According to Bayes' rule

The DDPM authors decided to condition this on x_0 (trick)

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

So, VLB becomes

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

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

For, $T = 4$ $$\sum_{t=2}^{4} \log\left(\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}\right) = \log\left(\prod_{t=2}^{4} \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}\right) = \log\left(\frac{\boxed{q(x_2|x_0)q(x_3|x_0)}q(x_4|x_0)}{q(x_1|x_0)\boxed{q(x_2|x_0)q(x_3|x_0)}}\right)$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (6/15)

Eventually, we obtain

$$\sum_{t=2}^{T} \log\left(\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}\right) = \log\left(\frac{q(x_T|x_0)}{q(x_1|x_0)}\right)$$

Hence, VLB becomes

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

Log division rule gives us

$$\log\left(q(x_T|x_0)\right) \boxed{-\log\left(q(x_1|x_0)\right)} \boxed{+\log\left(q(x_1|x_0)\right)} - \log\left(p_\theta(x_0|x_1)\right)$$

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

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (7/15)

Simplification results in

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

$$\boxed{D_{KL}(q(x_T|x_0)||p(x_T))}+\sum_{t=2}^{T}D_{KL}(\boxed{q(x_{t-1}|x_t,x_0)}||\boxed{p_\theta(x_{t-1}|x_t)})-\log(p_\theta(x_0|x_1))$$

This part has no learnable parameters

$$\mathcal{N}(x_{t-1};\ \tilde{\mu}_t(x_t,x_0),\tilde{\beta}_t\mathbf{I})$$

$$\mathcal{N}(x_{t-1};\ \mu_\theta(x_t,t),\beta_t\mathbf{I})$$

$$q(x_{t-1}|x_t,x_0)=\frac{q(x_t|x_{t-1},x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}$$ Bayes' rule

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (8/15)

$$\propto \exp\left(-\frac{1}{2}\left(\frac{\left(x_t-\sqrt{\alpha_t}x_{t-1}\right)^2}{\beta_t}+\frac{\left(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0\right)^2}{1-\bar{\alpha}_{t-1}}-\frac{\left(x_t-\sqrt{\bar{\alpha}_t}x_0\right)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$=\exp\left(-\frac{1}{2}\left(\frac{x_t^2-2\sqrt{\alpha_t}x_t x_{t-1}+\alpha_t x_{t-1}^2}{\beta_t}+\frac{x_{t-1}^2-2\sqrt{\bar{\alpha}_{t-1}}x_0 x_{t-1}+\bar{\alpha}_{t-1}x_0^2}{1-\bar{\alpha}_{t-1}}-\frac{\left(x_t-\sqrt{\bar{\alpha}_t}x_0\right)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$=\exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2-\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1}+C(x_t,x_0)\right)\right)$$

$$\beta_t = 1-\alpha_t$$

$$\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$$

$$\tilde{\beta}_t = \frac{1}{\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}} = \frac{\beta_t(1-\bar{\alpha}_{t-1})}{\alpha_t(1-\bar{\alpha}_{t-1})+\beta_t} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\bar{\alpha}_{t-1} = \alpha_1.\alpha_2.\alpha_3 \ldots \alpha_{t-1}$$

$$\bar{\alpha}_t = \alpha_1.\alpha_2.\alpha_3 \ldots \alpha_{t-1}\alpha_t$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (9/15)

$$\exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)x_{t-1}^2-\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1}+C(x_t,x_0)\right)\right)$$

$$\tilde{\mu}_t(x_t,x_0)=\frac{\frac{\sqrt{\alpha_t}}{\beta_t}x_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0}{\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}}=\frac{\sqrt{\alpha_t}}{\beta_t}x_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\left(\frac{1}{\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}}\right)$$

$$=\frac{\sqrt{\alpha_t}}{\beta_t}x_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right)$$

$$=\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t+\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (10/15)

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0$$

From forward process → $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon$

$$\Rightarrow x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon\right)$$

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon\right)$$

$$= \left[\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}\right] x_t - \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}\varepsilon$$

$$= Ax_t - B\varepsilon$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (11/15)

$$A = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)} = \frac{\sqrt{\bar{\alpha}_t}\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)} \qquad \beta_t = 1-\alpha_t$$

$$= \frac{\sqrt{\alpha_t\bar{\alpha}_{t-1}}\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)} \qquad \bar{\alpha}_t = \alpha_t\bar{\alpha}_{t-1}$$

$$= \frac{\boxed{\alpha_t\sqrt{\bar{\alpha}_{t-1}}} - \alpha_t\sqrt{\bar{\alpha}_{t-1}}\bar{\alpha}_{t-1} + \sqrt{\bar{\alpha}_{t-1}} - \boxed{\alpha_t\sqrt{\bar{\alpha}_{t-1}}}}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}\boxed{(1-\bar{\alpha}_t)}}{\sqrt{\bar{\alpha}_t}\boxed{(1-\bar{\alpha}_t)}} \qquad \bar{\alpha}_t = \alpha_t\bar{\alpha}_{t-1}$$

$$= \frac{\boxed{\sqrt{\bar{\alpha}_{t-1}}}}{\sqrt{\alpha_t}\boxed{\sqrt{\bar{\alpha}_{t-1}}}} \qquad \bar{\alpha}_t = \alpha_t\bar{\alpha}_{t-1}$$

$$= \frac{1}{\sqrt{\alpha_t}}$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (12/15)

$$B = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_t)}$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}\left(\sqrt{1-\bar{\alpha}_t}\right)^2}$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\bar{\alpha}_t}\sqrt{1-\bar{\alpha}_t}}$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{\alpha_t\bar{\alpha}_{t-1}}\sqrt{1-\bar{\alpha}_t}} \qquad \bar{\alpha}_t = \alpha_t\bar{\alpha}_{t-1}$$

$$= \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}$$

$$\tilde{\mu}_t(x_t, x_0) = Ax_t - B\varepsilon$$

$$A = \frac{1}{\sqrt{\alpha_t}} \qquad B = \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}$$

$$\tilde{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}x_t - \frac{\beta_t}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}\varepsilon$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right)$$

Mean of the forward process posterior

$\mu_\theta(x_t, t)$ → Mean predicted by the neural network

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (13/15)

The distance between $\tilde{\mu}_t(x_t, x_0)$ and $\mu_\theta(x_t, t)$ is approximated using a *mean-squared error (MSE)*

$$L_t = \frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2$$

$$= \frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon \right) - \mu_\theta(x_t, t) \right\|^2$$

$\mu_\theta(x_t, t)$ can be represented the same way as $\tilde{\mu}_t(x_t, x_0)$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(x_t, t) \right)$$

$$L_t = \frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon \right) - \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(x_t, t) \right) \right\|^2$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (14/15)

Simplification results in

$$L_t = \boxed{\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)}} \|\varepsilon - \varepsilon_\theta(x_t, t)\|^2$$

Scaling term

Actual noise

Predicted noise

DDPM authors found out that ignoring this term gives better training results

$$L_t = \|\varepsilon - \varepsilon_\theta(x_t, t)\|^2$$

$$= \left\|\varepsilon - \varepsilon_\theta\left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon, t\right)\right\|^2$$

Optimized during training

Going back to the beginning $p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \boxed{\mu_\theta(x_t, t)}, \boxed{\Sigma_\theta(x_t, t)}\right)$

$$\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(x_t, t)\right)$$

$$\Sigma_\theta(x_t, t) = \sigma_t^2 \mathbf{I}$$

Source: https://arxiv.org/pdf/2006.11239.pdf

Reverse (Denoising) Process (15/15)

Reparameterization trick: $\mathcal{N}(\mu, \sigma^2) = \mu + \sigma \odot z$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)\right) + \sigma_t z \qquad t > 1$$

One final term remains: $\log\big(p_\theta(x_0|x_1)\big)$

$$\mathcal{N}(x_0;\ \mu_\theta(x_1, 1), \sigma_1^2\mathbf{I})$$

The authors decided to sample this term noiselessly $z = 0$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)\right) \qquad t = 1$$

Source: https://arxiv.org/pdf/2006.11239.pdf

DDPM Training Process

Source image x_0 is randomly sampled from the original data distribution $q(x_0)$

t (time step) is sampled uniformly between 1 and T

Actual noise ε is sampled from a normal distribution

A neural network is trained via gradient descent to predict the noise ε_θ

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

$$L_t = \left\| \varepsilon - \varepsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t\right) \right\|^2$$

Source: https://arxiv.org/pdf/2006.11239.pdf

DDPM Sampling Process

Sampling starts from x_T which is an *isotropic Gaussian distribution*

Neural network gradually denoises it until $t = 1$

A slightly less denoised image x_{t-1} can be obtained using this equation

An image x_0 is returned that looks similar to the original data distribution

Algorithm 2 Sampling

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

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(x_t, t) \right) + \sigma_t z$$

$$\beta_t = 1 - \alpha_t$$

Source: https://arxiv.org/pdf/2006.11239.pdf

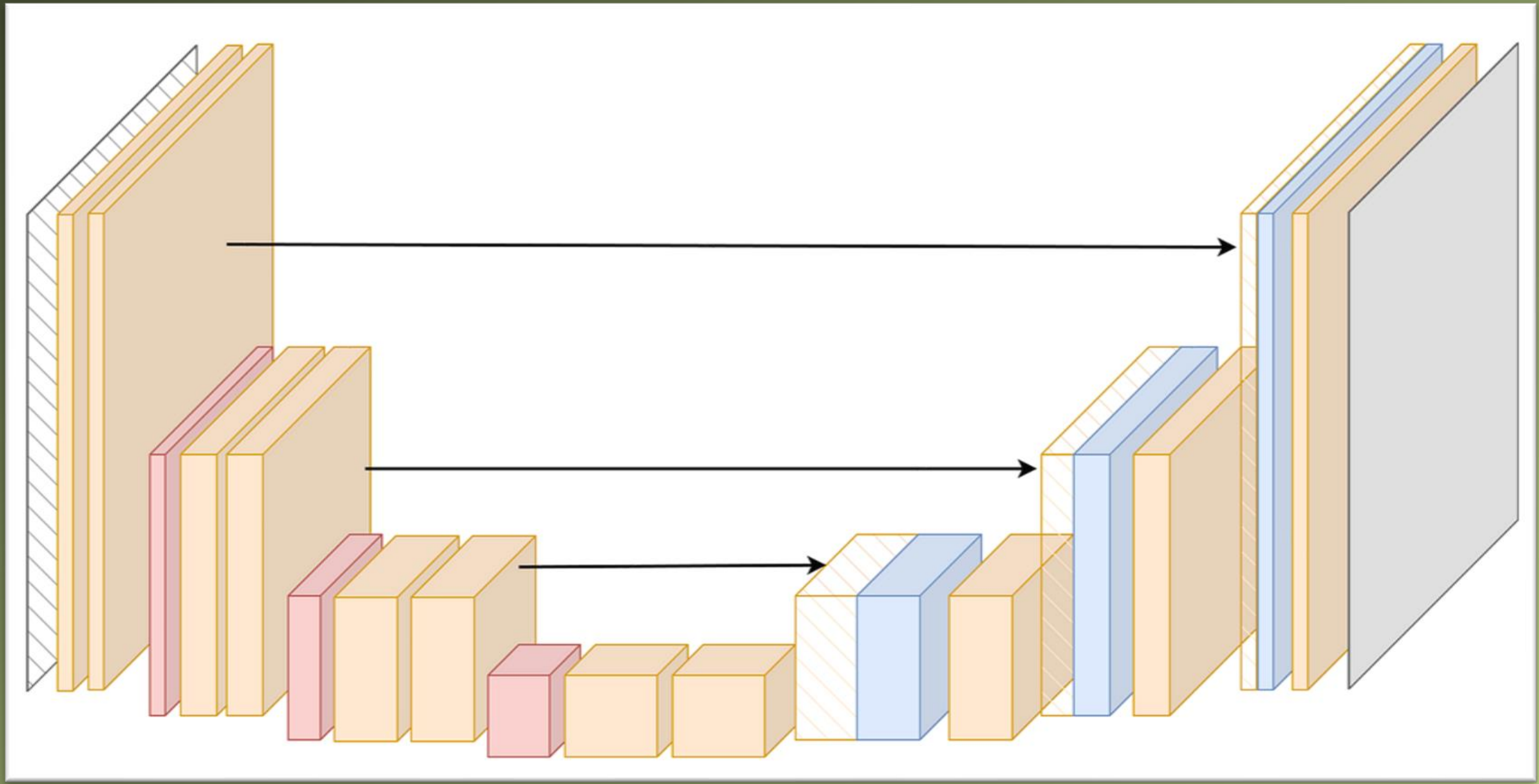
Neural Network (U-Net) Architecture

Position embeddings → To operate on a particular noise level

ResNet blocks → To use skip connections for merging the output of a previous layer with a layer ahead

Attention module → To allow a neural network to focus on a particular information at a time and ignore the rest

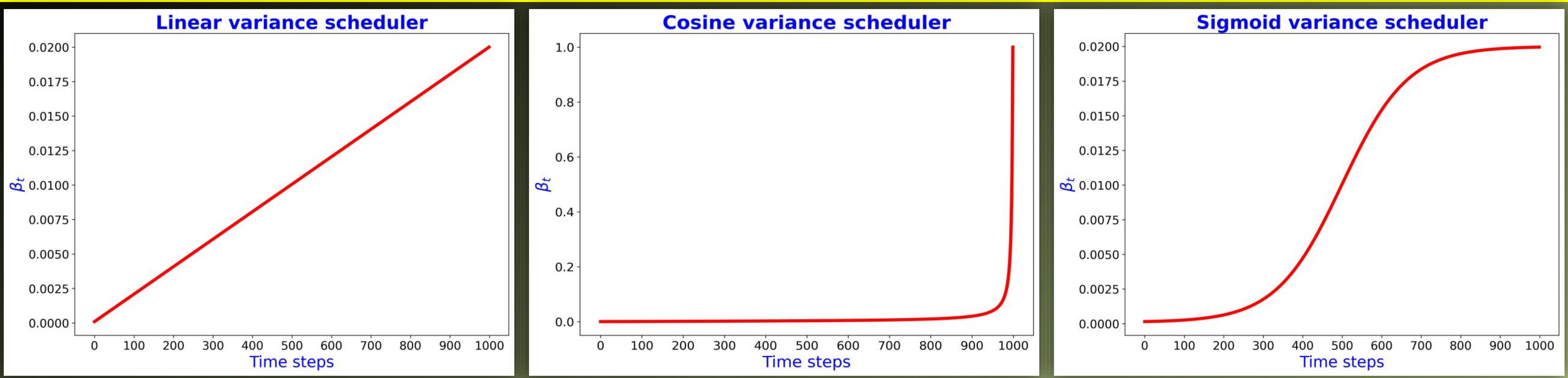
GroupNorm → To divide channels into groups and normalize features within each group. It works well for models with small batch sizes



A typical U-Net architecture

Source: https://towardsdatascience.com/u-net-explained-understanding-its-image-segmentation-architecture-56e4842e313a

Different Variance Schedulers (1/2)

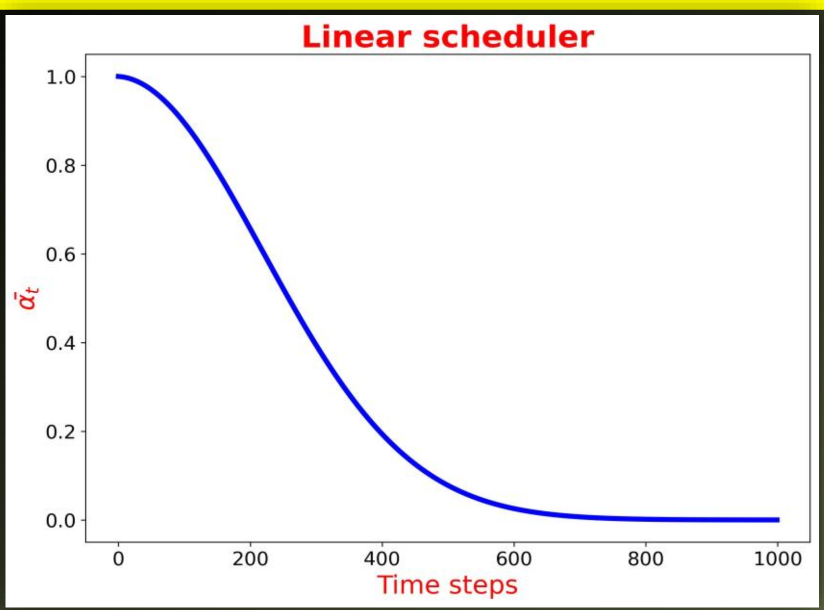


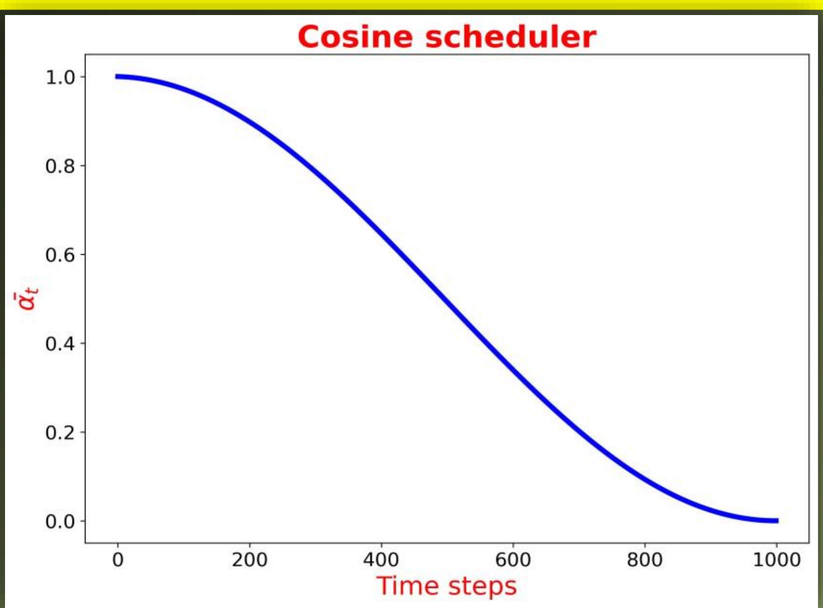
Linear

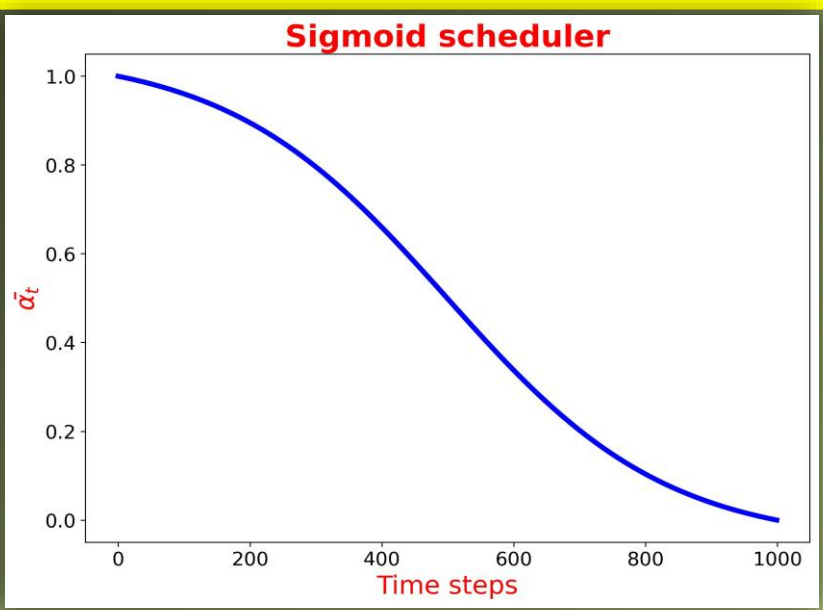
Cosine

Source: https://arxiv.org/pdf/2102.09672.pdf

Different Variance Schedulers (2/2)







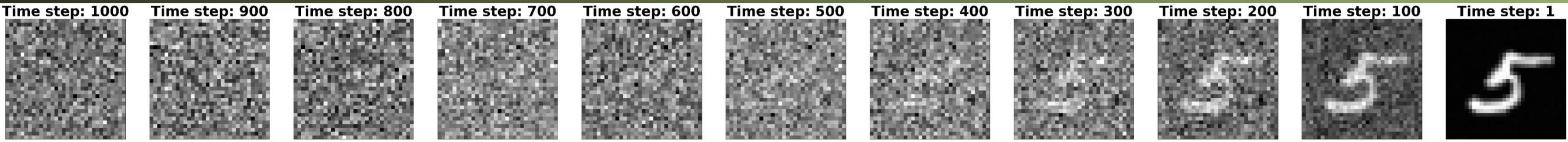
$\bar{\alpha}_t$ plot for *linear*, *cosine* and *sigmoid* scheduler

These plots show how quickly or slowly the information in the source image is destroyed

We can easily observe that the information is destroyed much quicker in the case of *linear* than in cosine and sigmoid schedulers

Source: https://arxiv.org/pdf/2102.09672.pdf

Reverse Process Output Using Linear Scheduler

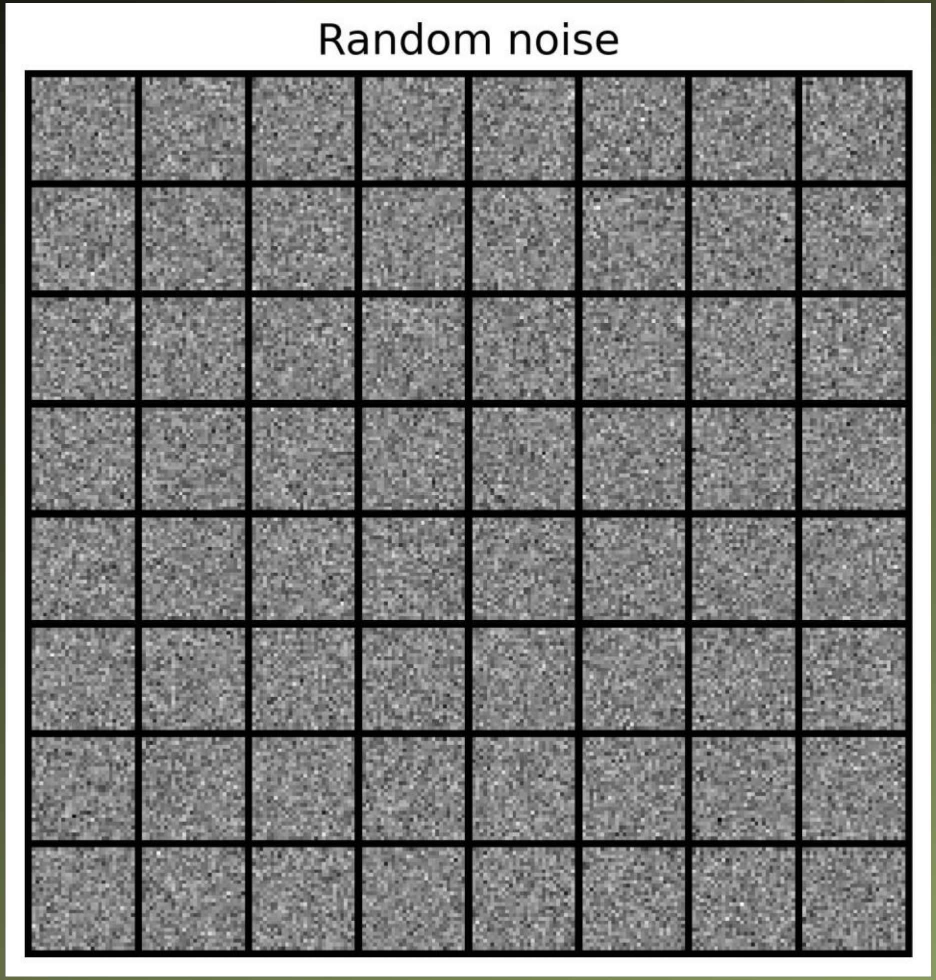


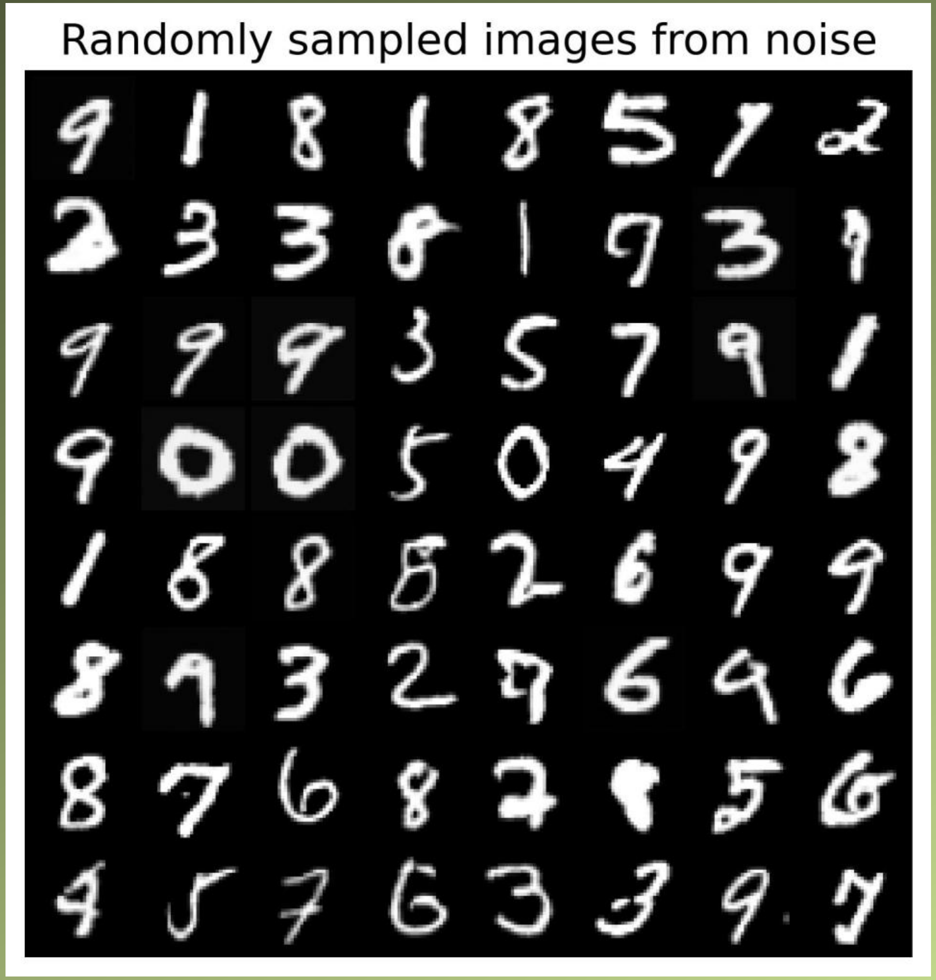
Reverse (denoising) process output

The final output should look like it came from the original data distribution

Source: https://arxiv.org/pdf/2006.11239.pdf

Random Sampling Using Linear Scheduler





DDPM models will be able to generate actual images from noise only if trained well

Source: https://arxiv.org/pdf/2006.11239.pdf

References

- ***The annotated diffusion model:*** https://huggingface.co/blog/annotated-diffusion
- ***What are diffusion models:*** https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- ***Denoising Diffusion Probabilistic Models:*** https://arxiv.org/pdf/2006.11239.pdf
- ***Improved Denoising Diffusion Probabilistic Models:*** https://arxiv.org/pdf/2102.09672.pdf
- ***U-Net Architecture:*** https://towardsdatascience.com/u-net-explained-understanding-its-image-segmentation-architecture-56e4842e313a

Source: https://arxiv.org/pdf/2006.11239.pdf

THE END