Diffusion Models From Scratch

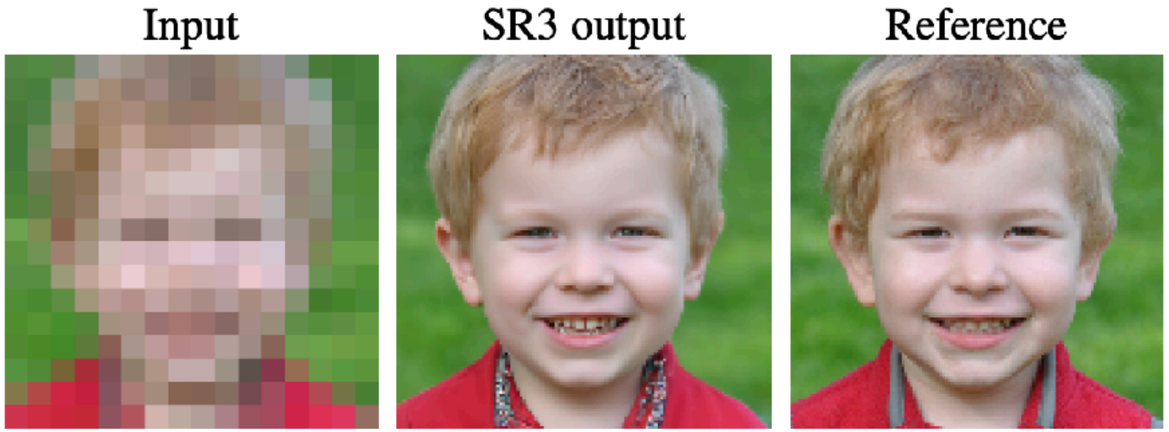
Gavin Kerrigan
CS 274E: Fall 2024
gavin.k@uci.edu
GavinKerrigan.github.io

Why Diffusion Models?

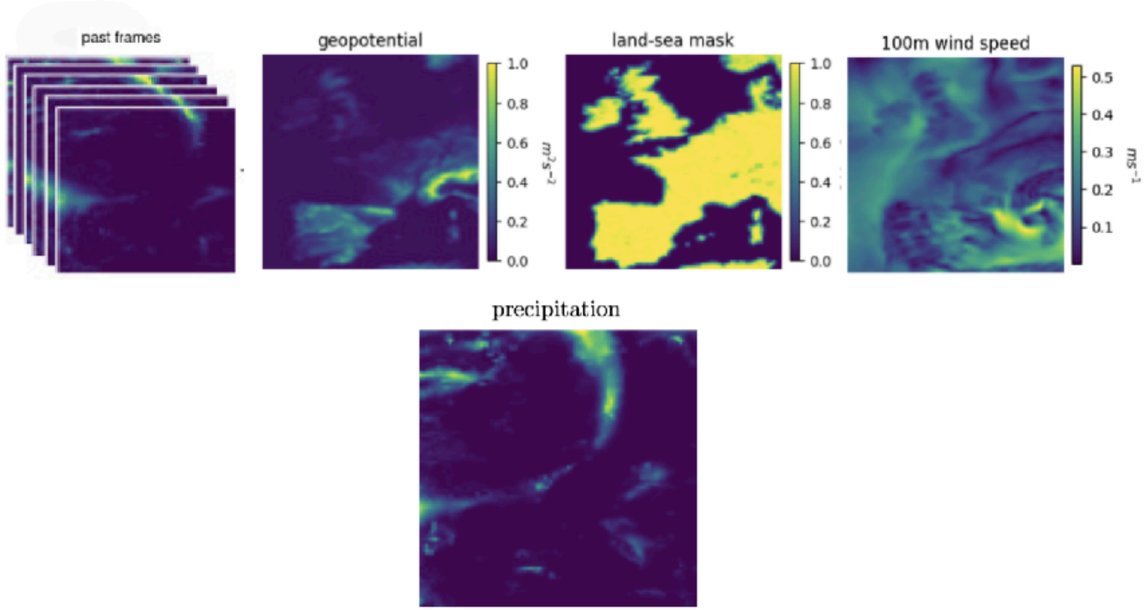
“An ornate oil painting of a cat wearing a wizard hat”
[DALL-E 2, OpenAI, 2022]

“A movie trailer … of the space man wearing a red wool knitted motorcycle helmet, blue sky, salt desert, …”
[Sora, OpenAI, 2024]

Why Diffusion Models?

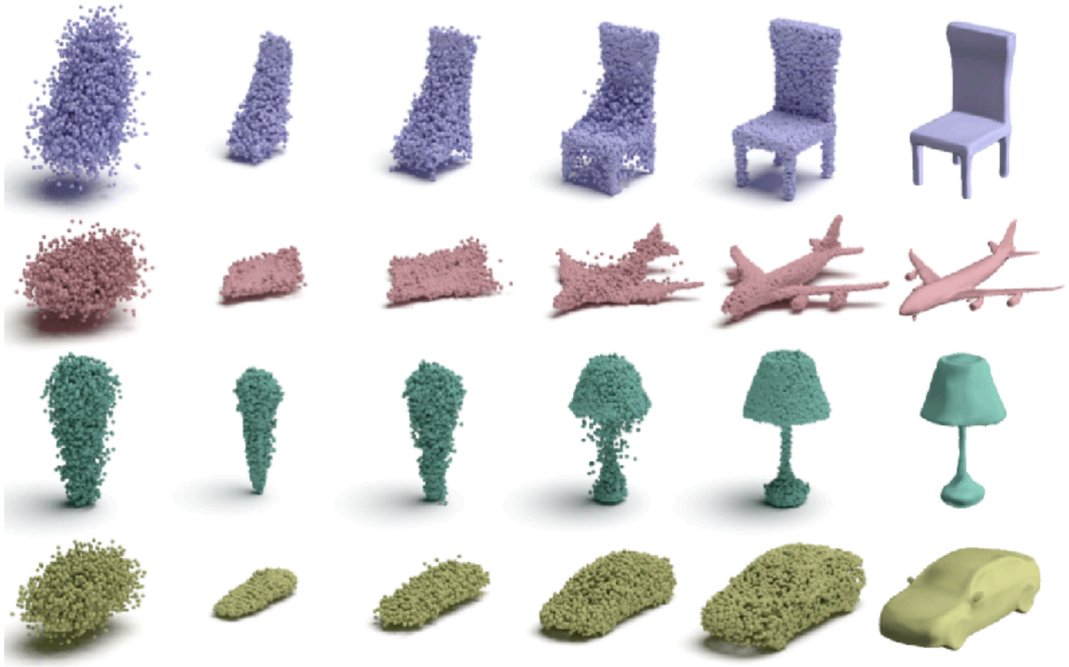


[Saharia et al., 2021]

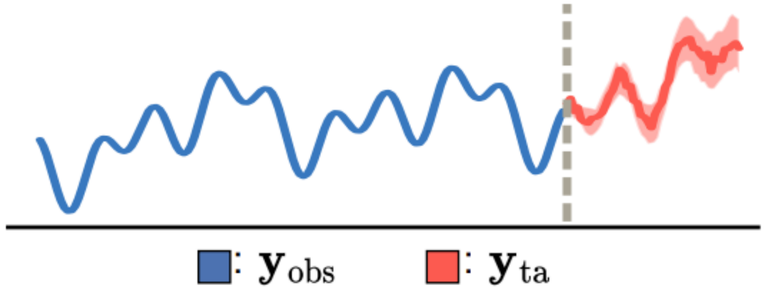


[Asperti et al., 2023]

Why Diffusion Models?



[Cai et al., 2020]



[Kollovieh et al., 2023]

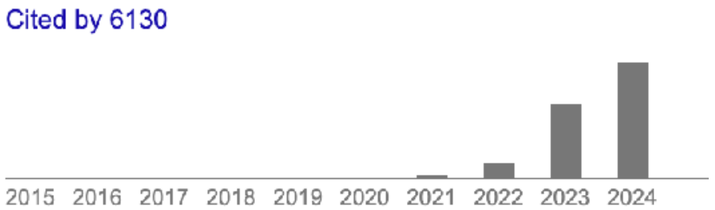
Diffusion Generative Models

Diffusion models are a class of deep generative models
that generate data by iterative denoising

Some history:

- Introduced in 2015 [Sohl-Dickstein et al., 2015], inspired by non-equilibrium thermodynamics
- Practical improvements lead to SOTA results [Ho et al., 2020]
- Continuous time, score-based models [Song et al., 2021]

Citations Counts of [Sohl-Dickstein et al., 2015]



Denoising Diffusion Probabilistic Models (DDPM)
[Ho et al., 2020]

“Creating noise from data is easy; creating data from noise is generative modeling.”
[Song et al., 2021]

Denoising Diffusion Probabilistic Models (DDPM)
[Ho et al., 2020]

"Creating noise from data is easy; creating data from noise is generative modeling."
[Song et al., 2021]

Creating Noise from Data

Have data samples $x_0 \sim q(x_0)$

Creating Noise from Data

Have data samples $x_0 \sim q(x_0)$

Fix some integer T

- Typically large, $T \approx 1000$

Creating Noise from Data

Have data samples $x_0 \sim q(x_0)$

Fix some integer T

Typically large, $T \approx 1000$

Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

Creating Noise from Data

Have data samples $x_0 \sim q(x_0)$

Fix some integer T

Typically large, $T \approx 1000$

Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

"Forward" Diffusion Process:

Slowly make data noisier over T discrete steps:

Creating Noise from Data

Have data samples $x_0 \sim q(x_0)$

Fix some integer T

Typically large, $T \approx 1000$

Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

"Forward" Diffusion Process:

Slowly make data noisier over T discrete steps:

$$x_1 = \sqrt{1-\beta_1}x_0 + \sqrt{\beta_1}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$

Creating Noise from Data

Have data samples $x_0 \sim q(x_0)$

Fix some integer T

Typically large, $T \approx 1000$

Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

"Forward" Diffusion Process:

Slowly make data noisier over T discrete steps:

$$x_1 = \sqrt{1-\beta_1}x_0 + \sqrt{\beta_1}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$

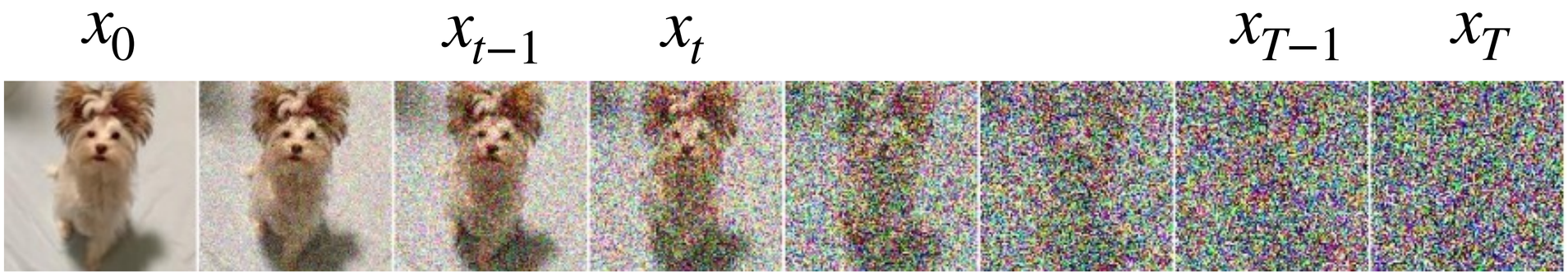
$$\bullet \bullet \bullet$$

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$

Creating Noise from Data

"Forward" Diffusion Process:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$



Creating Noise from Data

"Forward" Diffusion Process:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$

Defines forward transition densities:

$$x_t \sim q(x_t \mid x_{t-1})$$
$$= \mathcal{N}\left(x_t \mid \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$

Joint distribution over noisy datapoints:

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$ without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t} (1 - \beta_s)$

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$ without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t} (1 - \beta_s)$

$$x_t = \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon_t$$

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$ without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t} (1 - \beta_s)$

$$x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\,\epsilon_t$$

Plug-in

$$= \sqrt{1-\beta_t}\left(\sqrt{1-\beta_{t-1}}\,x_{t-2} + \sqrt{\beta_{t-1}}\,\epsilon_{t-1}\right) + \sqrt{\beta_t}\,\epsilon_t$$

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$ without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t} (1 - \beta_s)$

$$x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t}\epsilon_t$$

$$= \boxed{\sqrt{1-\beta_t}}\left(\sqrt{1-\beta_{t-1}} x_{t-2} + \boxed{\sqrt{\beta_{t-1}}\epsilon_{t-1}}\right) + \boxed{\sqrt{\beta_t}\epsilon_t}$$

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

Sum of independent Gaussians

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$

$$
\begin{aligned}
x_t &= \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_t \\
&= \sqrt{1-\beta_t}\left(\sqrt{1-\beta_{t-1}}x_{t-2} + \sqrt{\beta_{t-1}}\epsilon_{t-1}\right) + \sqrt{\beta_t}\epsilon_t \\
&= \sqrt{(1-\beta_t)(1-\beta_{t-1})}x_{t-2} + \sqrt{1-(1-\beta_t)(1-\beta_{t-1})}\epsilon \\
&\cdots \\
&= \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)
\end{aligned}
$$

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$ without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t} (1 - \beta_s)$

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\alpha_t} x_0 + (1 - \alpha_t) I\right)$$

Creating Noise from Data

Step-by-Step simulation can be slow; requires $O(t)$ calculations

Can directly sample $x_t \mid x_0$ without needing intermediate values

Define $\alpha_t = \prod_{s=1}^{t} (1 - \beta_s)$

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\alpha_t} x_0 + (1 - \alpha_t) I\right)$$

As long as $\alpha_T \approx 0$, ending distribution is

$$q(x_T \mid x_0) \approx \mathcal{N}(x_T \mid 0, I)$$

Creating Noise from Data

Marginal distribution at time t?

$$q(x_t) = \int q(x_t \mid x_0) q(x_0) \, \mathrm{d}x_0$$

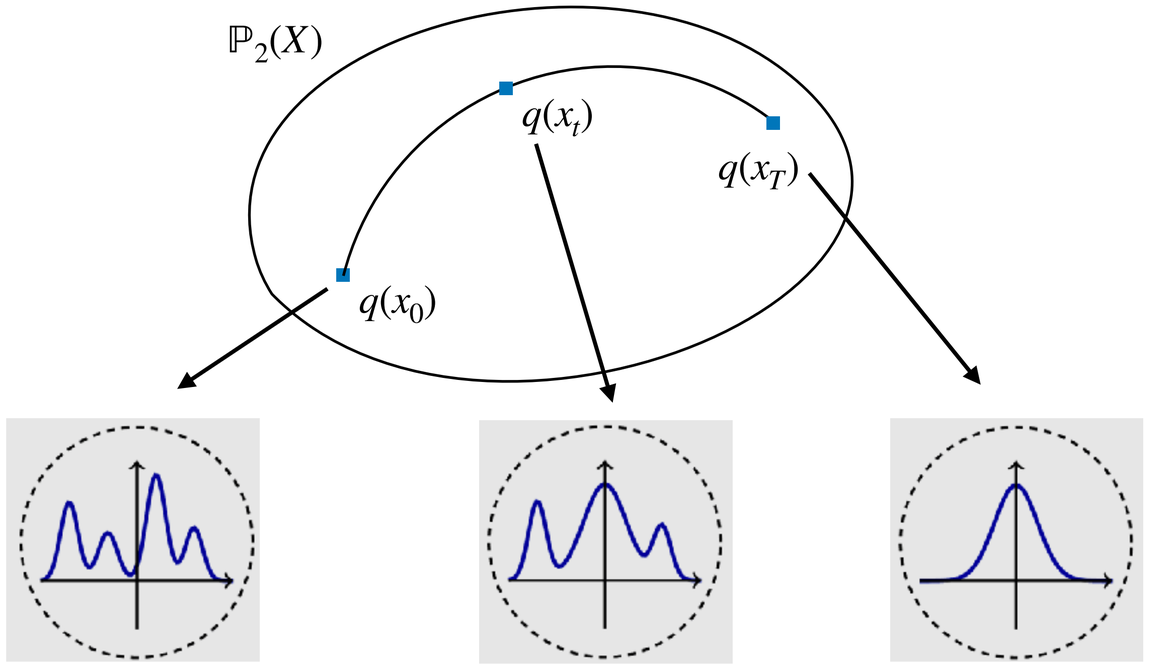
$$q(x_T \mid x_0) \approx \mathcal{N}(x_T \mid 0, I) \qquad q(x_T) \approx \mathcal{N}(0, I)$$

Creating Noise from Data

Marginal distribution at time t?

$$q(x_t) = \int q(x_t \mid x_0) q(x_0)\, \mathrm{d}x_0$$

$$q(x_T \mid x_0) \approx \mathcal{N}(x_T \mid 0, I) \qquad q(x_T) \approx \mathcal{N}(0, I)$$



Recap

1. Define a forward process

Fix number of steps T

Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

Construct Gaussian forward transitions: $x_t \sim q(x_t \mid x_{t-1})$

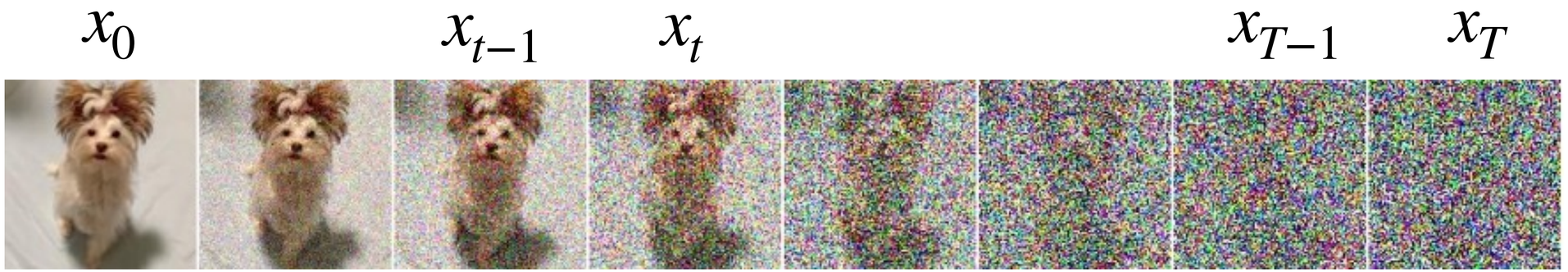
Can sample in one step: $x_t \sim q(x_t \mid x_0)$

Questions?

“Creating noise from data is easy; creating data from noise is generative modeling.”
[Song et al., 2021]

Creating Data from Noise

What if we could run our process backwards in time?



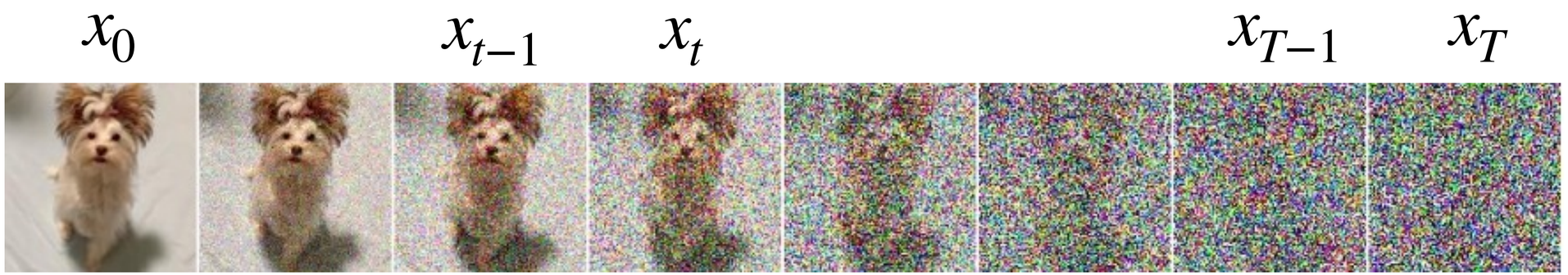
Creating Data from Noise

What if we could run our process backwards in time?

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

For $t = T, T-1, \ldots, 1$:

$$x_{t-1} \sim q(x_{t-1} \mid x_t)$$



Creating Data from Noise

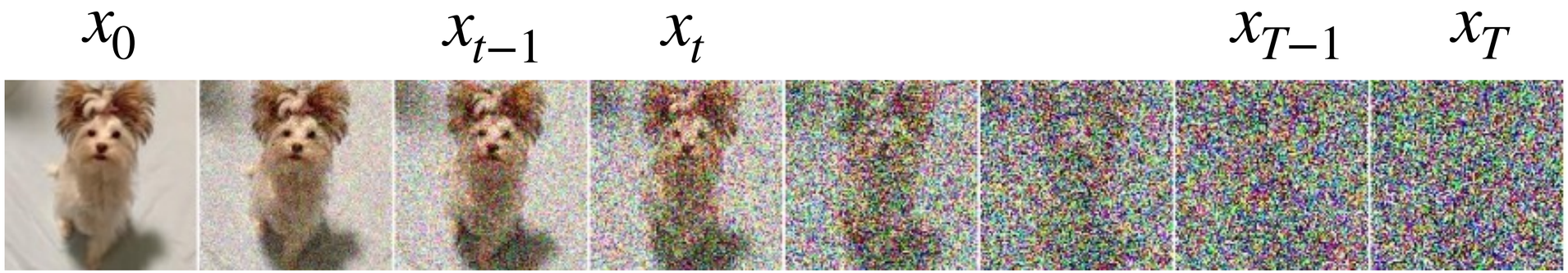
What if we could run our process backwards in time?

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

For $t = T, T-1, \ldots, 1$:

$$x_{t-1} \sim q(x_{t-1} \mid x_t)$$

Produces $x_0 \sim q(x_0)$ as our path $q(x_t)$ approximately interpolates $q(x_T)$ and $q(x_0)$



Creating Data from Noise

Requires sampling from the reverse transition densities

$$x_{t-1} \sim q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})q(x_{t-1})}{q(x_t)}$$

Creating Data from Noise

Requires sampling from the reverse transition densities

$$x_{t-1} \sim q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})q(x_{t-1})}{q(x_t)}$$

Know forward transitions $q(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\alpha_t}x_0 + (1-\alpha_t)I\right)$

Creating Data from Noise

Requires sampling from the reverse transition densities

$$x_{t-1} \sim q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})q(x_{t-1})}{q(x_t)}$$

Know forward transitions $q(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\alpha_t}x_0 + (1-\alpha_t)I\right)$

Marginals are intractable $q(x_t) = \int q(x_t \mid x_0)q(x_0)\,\mathrm{d}x_0$

Creating Data from Noise

Requires sampling from the reverse transition densities

$$x_{t-1} \sim q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})q(x_{t-1})}{q(x_t)}$$

Let's do variational inference!

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t)$$

Creating Data from Noise

Requires sampling from the reverse transition densities

$$x_{t-1} \sim q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1})q(x_{t-1})}{q(x_t)}$$

Let's do variational inference!

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t)$$

Choose a Gaussian parametrization

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

Creating Data from Noise

How can we train such a model?

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

Creating Data from Noise

How can we train such a model?

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

Defines a distribution over x_0:

$$p_\theta(x_0) = q(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

Creating Data from Noise

How can we train such a model?

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

Defines a distribution over x_0:

$$p_\theta(x_0) = q(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

Want to maximize likelihood of data x_0

$$\theta^* = \text{argmax}_\theta \mathbb{E}_{x_0 \sim q(x_0)} \log p_\theta(x_0)$$

Variational Inference Reminders

Want to maximize likelihood of data x_0

Encoder: $q(z \mid x_0)$ Decoder: $p_\theta(x_0 \mid z)$

Variational Inference Reminders

Want to maximize likelihood of data x_0

Encoder: $q(z \mid x_0)$ Decoder: $p_\theta(x_0 \mid z)$

Likelihood is intractable: $p_\theta(x_0) = \int p_\theta(x \mid z)p(z)dz$

Variational Inference Reminders

Want to maximize likelihood of data x_0

Encoder: $q(z \mid x_0)$ Decoder: $p_\theta(x_0 \mid z)$

Likelihood is intractable: $p_\theta(x_0) = \int p_\theta(x \mid z) p(z) dz$

Instead optimize the ELBO: $\log p_\theta(x_0) \geq \mathbb{E}_{z \sim q(z|x_0)} \left[\log \frac{p_\theta(x_0, z)}{q(z \mid x)} \right]$

Creating Data from Noise

Likelihood is intractable:

$$p_\theta(x_0) = q(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

… use the ELBO!

Creating Data from Noise

Likelihood is intractable:

$$p_\theta(x_0) = q(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

… use the ELBO!

Treating x_0 as data and $x_{1:T}$ as latent variables, the usual ELBO (e.g., in a VAE) is

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_{1:T} \sim q(x_{1:T} \mid x_0)} \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} \right]$$

ELBO Analysis

Let’s analyze the ELBO further to simplify things

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

ELBO Analysis

Let's analyze the ELBO further to simplify things

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

$$= \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0)} \left[\log \frac{p_\theta(x_0 \mid x_{1:T}) p_\theta(x_{1:T})}{q(x_{1:T} \mid x_0)} \right]$$

ELBO Analysis

Let's analyze the ELBO further to simplify things

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

$$= \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0 \mid x_{1:T})p_\theta(x_{1:T})}{q(x_{1:T} \mid x_0)}\right]$$

Process is Markov

$$= \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0 \mid x_1)p_\theta(x_{1:T})}{q(x_{1:T} \mid x_0)}\right]$$

ELBO Analysis

Let's analyze the ELBO further to simplify things

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

$$= \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0 \mid x_{1:T})p_\theta(x_{1:T})}{q(x_{1:T} \mid x_0)}\right]$$

$$= \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0 \mid x_1)p_\theta(x_{1:T})}{q(x_{1:T} \mid x_0)}\right]$$

$$= \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0 \mid x_1)\right] - \mathrm{KL}\left[q(x_{1:T} \mid x_0) \,||\, p_\theta(x_{1:T})\right]$$

ELBO Analysis

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_1 \sim q(x_1|x_0)} \left[\log p_\theta(x_0 \mid x_1)\right] - \mathrm{KL}\left[q(x_{1:T} \mid x_0) \mid\mid p_\theta(x_{1:T})\right]$$

Notice the KL involves multiple time steps simultaneously…

ELBO Analysis

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_1 \sim q(x_1|x_0)} \left[\log p_\theta(x_0 \mid x_1)\right] - \mathrm{KL}\left[q(x_{1:T} \mid x_0) \mid\mid p_\theta(x_{1:T})\right]$$

Notice the KL involves multiple time steps simultaneously…

KL Divergence Chain Rule:

$$\boxed{\mathrm{KL}[p(x,y) \mid\mid q(x,y)]} = \boxed{\mathrm{KL}[p(x) \mid\mid q(x)]} + \boxed{\mathbb{E}_{x \sim p(x)} \mathrm{KL}[p(y \mid x) \mid\mid q(y \mid x)]}$$

Joint KL = Marginal KL + Expected conditional KL

Proof follows directly from definition of KL

ELBO Analysis

KL Divergence Chain Rule:

$$\mathrm{KL}[p(x,y)\,||\,q(x,y)] = \mathrm{KL}[p(x)\,||\,q(x)] + \mathbb{E}_{x\sim p(x)}\mathrm{KL}[p(y\mid x)\,||\,q(y\mid x)]$$

Apply this to our ELBO to condition on x_T:

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0\mid x_1)\right] - \mathrm{KL}\left[q(x_{1:T}\mid x_0)\,||\,p_\theta(x_{1:T})\right]$$

ELBO Analysis

KL Divergence Chain Rule:

$$\mathrm{KL}[p(x,y)\,||\,q(x,y)] = \mathrm{KL}[p(x)\,||\,q(x)] + \mathbb{E}_{x\sim p(x)}\mathrm{KL}[p(y\mid x)\,||\,q(y\mid x)]$$

Apply this to our ELBO to condition on x_T:

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0\mid x_1)\right] - \mathrm{KL}\left[q(x_{1:T}\mid x_0)\,||\,p_\theta(x_{1:T})\right]$$

ELBO Analysis

KL Divergence Chain Rule:

$$\mathrm{KL}[p(x,y)\,||\,q(x,y)] = \mathrm{KL}[p(x)\,||\,q(x)] + \mathbb{E}_{x\sim p(x)}\mathrm{KL}[p(y\mid x)\,||\,q(y\mid x)]$$

Apply this to our ELBO to condition on x_T:

$$\begin{aligned}\log p_\theta(x_0) &\geq \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0\mid x_1)\right] - \mathrm{KL}\left[q(x_{1:T}\mid x_0)\,||\,p_\theta(x_{1:T})\right] \\ &= \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0\mid x_1)\right] - \mathrm{KL}[q(x_T\mid x_0)\,||\,p_\theta(x_T)] \\ &\quad - \mathbb{E}_q\mathrm{KL}[q(x_{1:T-1}\mid x_0, X_T)\,||\,p_\theta(x_{1:T-1}\mid x_T)]\end{aligned}$$

ELBO Analysis

KL Divergence Chain Rule:

$$\mathrm{KL}[p(x,y)\,||\,q(x,y)] = \mathrm{KL}[p(x)\,||\,q(x)] + \mathbb{E}_{x\sim p(x)}\mathrm{KL}[p(y\mid x)\,||\,q(y\mid x)]$$

…. now repeat:

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0\mid x_1)\right] - \mathrm{KL}[q(x_T\mid x_0)\,||\,p_\theta(x_T)]$$
$$- \mathbb{E}_q \mathrm{KL}[q(x_{1:T-1}\mid x_0, x_T)\,||\,p_\theta(x_{1:T-1}\mid x_T)]$$

ELBO Analysis

KL Divergence Chain Rule:

$$\mathrm{KL}[p(x,y)\,||\,q(x,y)] = \mathrm{KL}[p(x)\,||\,q(x)] + \mathbb{E}_{x\sim p(x)}\mathrm{KL}[p(y\mid x)\,||\,q(y\mid x)]$$

…. now repeat:

$$\log p_\theta(x_0) \geq \mathbb{E}_{x_1\sim q(x_1|x_0)}\left[\log p_\theta(x_0\mid x_1)\right] - \mathrm{KL}[q(x_T\mid x_0)\,||\,p_\theta(x_T)]$$
$$- \mathbb{E}_q\mathrm{KL}[q(x_{1:T-1}\mid x_0, x_T)\,||\,p_\theta(x_{1:T-1}\mid x_T)]$$

To eventually obtain:

$$\log p_\theta(x_0) \geq \mathbb{E}_q\Big[p_\theta(x_0\mid x_1) - \mathrm{KL}[q(x_T\mid x_0)\,||\,p_\theta(x_T)] - \sum_{t=2}^{T}\mathrm{KL}[q(x_{t-1}\mid x_t, x_0)\,||\,p_\theta(x_{t-1}\mid x_t)]\Big]$$

ELBO Analysis

$$\log p_\theta(x_0) \geq \mathbb{E}_q\left[p_\theta(x_0 \mid x_1) - \mathrm{KL}[q(x_T \mid x_0) \mid\mid p_\theta(x_T)] - \sum_{t=2}^{T} \mathrm{KL}[q(x_{t-1} \mid x_t, x_0) \mid\mid p_\theta(x_{t-1} \mid x_t)]\right]$$

Still very ugly! But we see three types of terms:

ELBO Analysis

$$\log p_\theta(x_0) \geq \mathbb{E}_q \left[p_\theta(x_0 \mid x_1) - \mathrm{KL}[q(x_T \mid x_0) \,||\, p_\theta(x_T)] - \sum_{t=2}^{T} \mathrm{KL}[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)] \right]$$

Still very ugly! But we see three types of terms:

"Decoder": $L_0 = \mathbb{E}_{x_1 \sim q(x_1|x_0)} p_\theta(x_0 \mid x_1)$

Analogous to the reconstruction term in a VAE

Assume Gaussian —> minimizing L_0 is equivalent to MSE regression on x_0

ELBO Analysis

$$\log p_\theta(x_0) \geq \mathbb{E}_q \left[p_\theta(x_0 \mid x_1) - \mathrm{KL}[q(x_T \mid x_0) \,||\, p_\theta(x_T)] - \sum_{t=2}^{T} \mathrm{KL}[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)] \right]$$

Still very ugly! But we see three types of terms:

"Decoder": $L_0 = \mathbb{E}_{x_1 \sim q(x_1|x_0)} p_\theta(x_0 \mid x_1)$

Analogous to the reconstruction term in a VAE

Assume Gaussian —> minimizing L_0 is equivalent to MSE regression on x_0

"Prior": $L_T = \mathbb{E}_{x_T \sim q(x_T|x_0)} \mathrm{KL}\left[q(x_T \mid x_0) \,||\, p_\theta(x_T) \right]$

Recall $q(x_T \mid x_0) \approx \mathcal{N}(0, I)$

Set $p_\theta(x_T) = \mathcal{N}(0, I)$ —> can ignore this loss term

ELBO Analysis

$$\log p_\theta(x_0) \geq \mathbb{E}_q \left[p_\theta(x_0 \mid x_1) - \mathrm{KL}[q(x_T \mid x_0) \,||\, p_\theta(x_T)] - \sum_{t=2}^{T} \mathrm{KL}[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)] \right]$$

Still very ugly! But we see three types of terms:

"Decoder": $L_0 = \mathbb{E}_{x_1 \sim q(x_1|x_0)} p_\theta(x_0 \mid x_1)$

Analogous to the reconstruction term in a VAE

Assume Gaussian —> minimizing L_0 is equivalent to MSE regression on x_0

"Prior": $L_T = \mathbb{E}_{x_T \sim q(x_T|x_0)} \mathrm{KL}\left[q(x_T \mid x_0) \,||\, p_\theta(x_T)\right]$

Recall $q(x_T \mid x_0) \approx \mathcal{N}(0, I)$

Set $p_\theta(x_T) = \mathcal{N}(0, I)$ —> can ignore this loss term

"Denoiser": $L_t = \mathbb{E}_{x_{t-1}, x_t \sim q(x_{t-1}, x_t|x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right]$

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_{x_{t-1}, x_t \sim q(x_{t-1}, x_t | x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t) \right]$$

Note this involves two distributions:

- True reverse distribution
- Model reverse distribution

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_{x_{t-1},x_t \sim q(x_{t-1},x_t|x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right]$$

Note this involves two distributions:

- True reverse distribution
- Model reverse distribution

True reverse distribution becomes tractable when conditioned on x_0!

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

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_{x_{t-1}, x_t \sim q(x_{t-1}, x_t | x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t) \right]$$

Note this involves two distributions:

- True reverse distribution
- Model reverse distribution

True reverse distribution becomes tractable when conditioned on x_0!

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

$$= \frac{q(x_t \mid x_{t-1}) q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}$$

ELBO Analysis

"Denoiser": $L_t = \mathbb{E}_{x_{t-1}, x_t \sim q(x_{t-1}, x_t | x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t) \right]$

Note this involves two distributions:

- True reverse distribution
- Model reverse distribution

True reverse distribution becomes tractable when conditioned on x_0!

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

$$= \frac{q(x_t \mid x_{t-1}) q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}$$

All components are Gaussian —> $q(x_{t-1} \mid x_t, x_0)$ is Gaussian

ELBO Analysis

After some tedious algebra:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1} \mid \mu_q(x_t, x_0), \sigma_q^2(t)I)$$

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}x_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t \qquad \sigma_q^2(t) = \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$$

Don't worry about the details. Just know:

- $q(x_{t-1} \mid x_t, x_0)$ is Gaussian
- We can easily compute its mean/variance

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_{x_{t-1},x_t \sim q(x_{t-1},x_t|x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right]$$

Note this involves two distributions:

- True reverse distribution (Gaussian, mean $\mu_q(x_t, x_0)$ and variance $\sigma_q^2(t)I$)
- Model reverse distribution

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_{x_{t-1}, x_t \sim q(x_{t-1}, x_t | x_0)} \text{KL} \left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t) \right]$$

Note this involves two distributions:

- True reverse distribution (Gaussian, mean $\mu_q(x_t, x_0)$ and variance $\sigma_q^2(t)I$)
- Model reverse distribution

Recall: we *choose* the model distribution to be Gaussian

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

For convenience, set $\Sigma_\theta(x_t, t) = \sigma_q^2(t)I$

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_{x_{t-1},x_t \sim q(x_{t-1},x_t|x_0)} \mathrm{KL}\left[q(x_{t-1} \mid x_t, x_0) \,||\, p_\theta(x_{t-1} \mid x_t)\right]$$

Note this involves two distributions:

- True reverse distribution (Gaussian, mean $\mu_q(x_t, x_0)$ and variance $\sigma_q^2(t)I$)
- Model reverse distribution

Recall: we *choose* the model distribution to be Gaussian

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

For convenience, set $\Sigma_\theta(x_t, t) = \sigma_q^2(t)I$

The KL between Gaussians has a closed form, so that

$$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} ||\mu_\theta(x_t, t) - \mu_q(x_t, x_0)||^2\right] + C$$

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} || \mu_\theta(x_t, t) - \mu_q(x_t, x_0) ||^2 \right]$$

How should we interpret this loss?

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} || \mu_\theta(x_t, t) - \boxed{\mu_q(x_t, x_0)} ||^2 \right]$$

How should we interpret this loss?

Recall $\boxed{\mu_q(x_t, x_0)}$ is the mean of $q(x_{t-1} \mid x_t, x_0)$

ELBO Analysis

"Denoiser": $$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} \left| \left| \mu_\theta(x_t, t) - \mu_q(x_t, x_0) \right| \right|^2 \right]$$

How should we interpret this loss?

Recall $\mu_q(x_t, x_0)$ is the mean of $q(x_{t-1} \mid x_t, x_0)$

Model $\mu_\theta(x_t, t)$ is a *denoiser* — best guess for x_{t-1} given only x_t

We are really training a model to denoise across many scales via MSE.

Model Parametrization

"Denoiser": $$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} || \mu_\theta(x_t, t) - \mu_q(x_t, x_0) ||^2 \right]$$

Can we simplify things even further?

Regression target has a lot of structure…

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} x_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$$

Model Parametrization

"Denoiser": $$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} || \mu_\theta(x_t, t) - \mu_q(x_t, x_0) ||^2 \right]$$

Can we simplify things even further?

Regression target has a lot of structure…

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} x_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$$

Model $\mu_\theta(x_t, t)$ already "knows" the coefficients and x_t

Model Parametrization

"Denoiser": $$L_t = \mathbb{E}_q \left[\frac{1}{2\sigma_q^2(t)} \left|\left| \mu_\theta(x_t, t) - \mu_q(x_t, x_0) \right|\right|^2 \right]$$

Can we simplify things even further?

Regression target has a lot of structure…

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} x_0 + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$$

Model $\mu_\theta(x_t, t)$ already "knows" the coefficients and x_t

Alternative parametrization:

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} \hat{x}_0^\theta(x_t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$$

i.e., model now predicts noise-free x_0 from x_t

Model Parametrization

Alternative parametrization:

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\hat{x}_0^\theta(x_t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t$$

$$L_t = C_t \mathbb{E}_{x_0, x_t}\left[||x_0 - \hat{x}_0{}^\theta(x_t, t)||^2 \right]$$

Model Parametrization

Alternative parametrization:

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\hat{x}_0^\theta(x_t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t$$

$$L_t = C_t \mathbb{E}_{x_0, x_t}\left[||x_0 - \hat{x}_0{}^\theta(x_t, t)||^2 \right]$$

Constant (often ignored in practice):

$$C_t = \frac{1}{2\sigma_q^2(t)} \frac{\alpha_{t-1}\beta_t^2}{(1-\alpha_t)^2}$$

Model Parametrization

Alternative parametrization:

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\hat{x}_0^\theta(x_t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t$$

$$L_t = C_t \mathbb{E}_{x_0, x_t}\left[||x_0 - \hat{x}_0{}^\theta(x_t, t)||^2 \right]$$

Constant (often ignored in practice):

$$C_t = \frac{1}{2\sigma_q^2(t)} \frac{\alpha_{t-1}\beta_t^2}{(1-\alpha_t)^2}$$

Note: many different parametrizations possible

- Directly predicting μ_q
- Predicting x_0
- Predicting ϵ
- ....

Putting Everything Together

1. Define a forward process

   Fix number of steps T

   Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

   Construct Gaussian forward transitions: $x_t \sim q(x_t \mid x_{t-1})$

Putting Everything Together

1. Define a forward process

   Fix number of steps T

   Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

   Construct Gaussian forward transitions: $x_t \sim q(x_t \mid x_{t-1})$

2. Learn the reverse process through variational inference

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t)$$

Putting Everything Together

1. Define a forward process

   Fix number of steps T

   Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

   Construct Gaussian forward transitions: $x_t \sim q(x_t \mid x_{t-1})$

2. Learn the reverse process through variational inference

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \sigma_q^2(t) I\right)$$

Putting Everything Together

1. Define a forward process

   Fix number of steps T

   Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

   Construct Gaussian forward transitions: $x_t \sim q(x_t \mid x_{t-1})$

2. Learn the reverse process through variational inference

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \sigma_q^2(t) I\right)$$

$\sigma_q^2(t)$: known, closed-form

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\hat{x}_0^\theta(x_t, t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t$$

Model $\hat{x}_0^\theta(x_t, t)$ predicts noise-free x_0 from x_t

Putting Everything Together

1. Define a forward process

   Fix number of steps T

   Choose a noise schedule $(\beta_t)_{t=1}^{T}$ of scalars

   Construct Gaussian forward transitions: $x_t \sim q(x_t \mid x_{t-1})$

2. Learn the reverse process through variational inference

$$q(x_{t-1} \mid x_t) \approx p_\theta(x_{t-1} \mid x_t)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1} \mid \mu_\theta(x_t, t), \sigma_q^2(t) I\right)$$

$\sigma_q^2(t)$: known, closed-form

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\hat{x}_0^\theta(x_t, t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t$$

Model $\hat{x}_0^\theta(x_t, t)$ predicts noise-free x_0 from x_t

Minimize:

$$L = \sum_{t=1}^{T} C_t \, \mathbb{E}_{x_0, x_t} \left|\left| x_0 - \hat{x}_0^\theta(x_t, t) \right|\right|^2$$

Diffusion Models Made Easy

Training Pseudocode:

1. Sample data $x_0 \sim q(x_0)$
2. Sample $t \sim \text{Unif}\{1,2,\ldots,T\}$
3. Sample noisy data $x_t \sim q(x_t \mid x_0)$
4. Predict noise-free version of x_t with a model $\hat{x}_0^\theta(x_t, t)$
5. Take a gradient step on the MSE $||x_0 - \hat{x}_0^\theta(x_t, t)||^2$

Sampling Pseudocode:

1. Sample pure noise $x_T \sim \mathcal{N}(0,I)$

For $t = T, \ldots, 2$:

2. Sample $x_{t-1} \sim p_\theta(x_{t-1} \mid x_t)$

3. Return $x_0 = \hat{x}_0^\theta(x_1, 1)$

To sample $x_{t-1} \sim p_\theta(x_{t-1} \mid x_t)$:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t), \sigma_q^2(t)I)$$

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\hat{x}_0^\theta(x_t, t) + \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}x_t$$

$$\sigma_q^2(t) = \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$$

DDPM Samples

Questions?

Practical Details

How do you choose the noise schedule (β_t)?

Many, many ways of doing this

Common choice: fix endpoints and linearly interpolate

e.g. original DDPM paper uses $\beta_1 = 10^{-4} \quad \beta_T = 0.02$

Practical Details

How do you choose the noise schedule (β_t)?

Many, many ways of doing this

Common choice: fix endpoints and linearly interpolate

e.g. original DDPM paper uses $\beta_1 = 10^{-4} \quad \beta_T = 0.02$

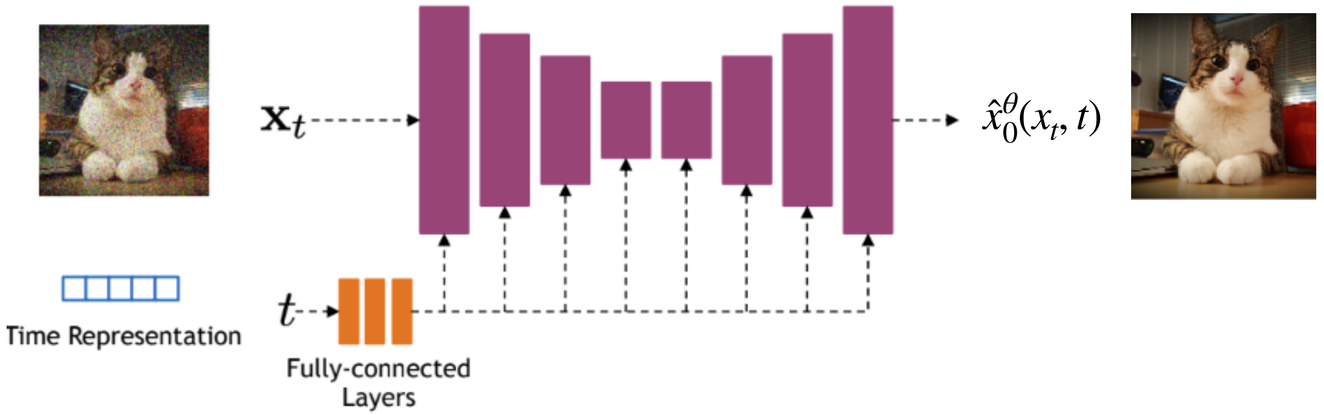
What model architecture should we use?

Many, many ways of doing this

Model $\hat{x}_0^\theta(x_t, t)$ predicts noise-free x_0 from x_t

Input: image; Output: image

Common choice: UNets, vision transformers



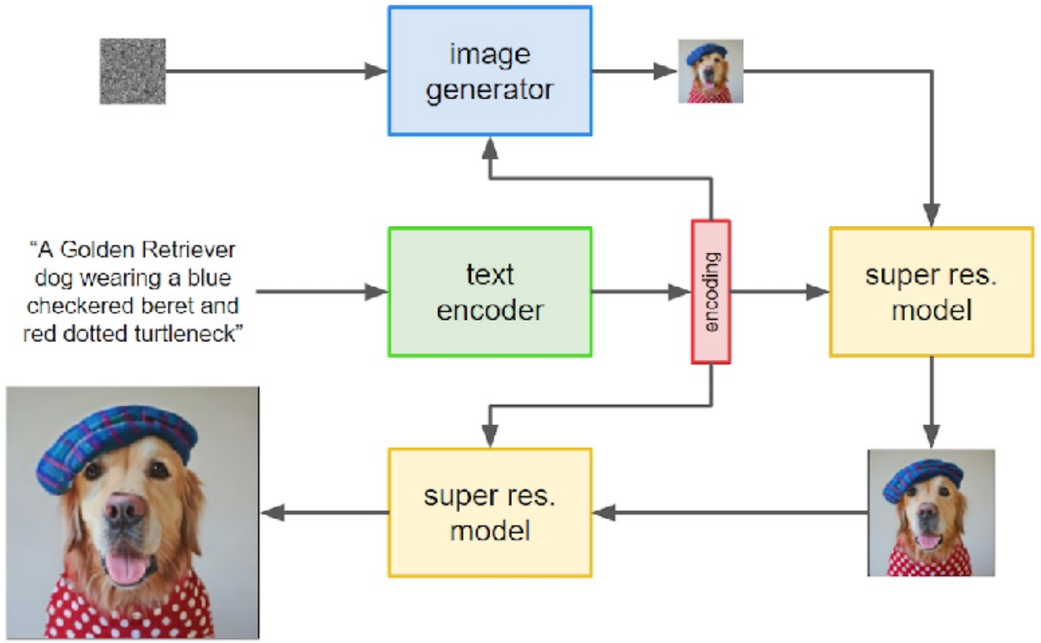
Practical Details

Can I train a conditional model?

Many, many ways of doing this

Basic approach: train model as described, but pass in y at training time

Challenge: having model make best use of conditioning information



Imagen, Google Brain, 2022

Practical Details

How do I speed up generation?

Sampling Pseudocode:

1. Sample pure noise $x_T \sim \mathcal{N}(0, I)$

For $t = T, \ldots, 2$:

2. Sample $x_{t-1} \sim p_\theta(x_{t-1} \mid x_t)$

3. Return $x_0 = \hat{x}_0^\theta(x_1, 1)$

Can require ~1000 model forward passes!

Many, many ways of doing this

- Distillation / Consistency Models
- Tricks to skip steps at sampling time (DDIM)

Where Do We Go From Here?

1. Continuous-Time perspectives

Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

Where Do We Go From Here?

1. Continuous-Time perspectives

2. Flow Matching & Stochastic Interpolants

Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

Preprint

FLOW MATCHING FOR GENERATIVE MODELING

Yaron Lipman[1,2] **Ricky T. Q. Chen**[1] **Heli Ben-Hamu**[2] **Maximilian Nickel**[1] **Matt Le**[1]
[1]Meta AI (FAIR) [2]Weizmann Institute of Science

Where Do We Go From Here?

1. Continuous-Time perspectives

Published as a conference paper at ICLR 2021

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

2. Flow Matching & Stochastic Interpolants

Preprint

FLOW MATCHING FOR GENERATIVE MODELING

Yaron Lipman[1,2] **Ricky T. Q. Chen**[1] **Heli Ben-Hamu**[2] **Maximilian Nickel**[1] **Matt Le**[1]
[1]Meta AI (FAIR) [2]Weizmann Institute of Science

3. Applications to non-image data
   - Video
   - Time series
   - Molecules
   - … text?

Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution

Aaron Lou [1] **Chenlin Meng** [1 2] **Stefano Ermon** [1]

Additional Resources

1. CVPR 2022 Tutorial

   cvpr2022-tutorial-diffusion-models.github.io

2. *Understanding Diffusion Models: A Unified Perspective*, Calvin Luo

   calvinyluo.com/2022/08/26/diffusion-tutorial.html

3. *What are Diffusion Models?,* Lilian Weng

   lilianweng.github.io/posts/2021-07-11-diffusion-models/

4. Reference Implementation

   github.com/lucidrains/denoising-diffusion-pytorch

Denoising Diffusion Probabilistic Models (DDPM)
[Ho et al., 2020]

“Creating noise from data is easy; creating data from noise is generative modeling.”
[Song et al., 2021]