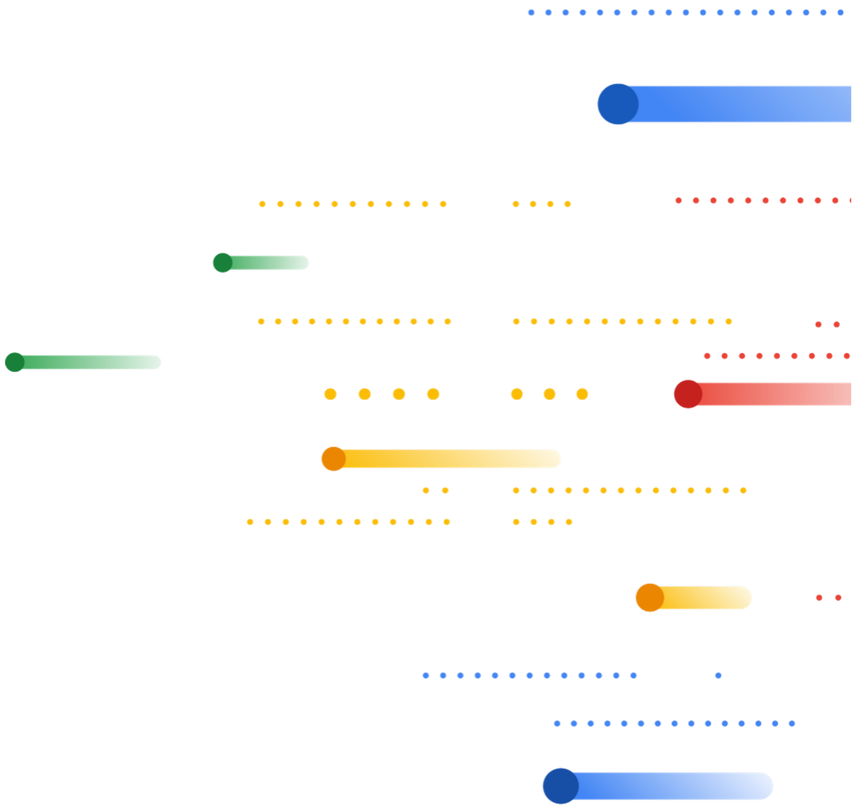
Slides from:

Google DeepMind

Variational Autoencoders and Diffusion Models



Ruiqi Gao @Stanford cs231n
May 25, 2023

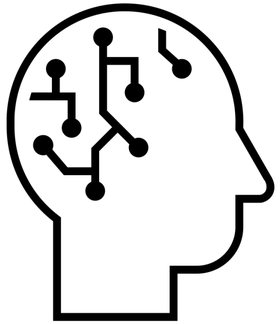
Deep Generative Models

Deep Generative Models

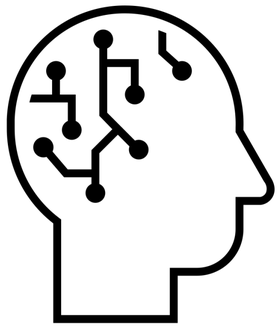
Learning to generate data

Samples from a Data Distribution

Train



Neural Network

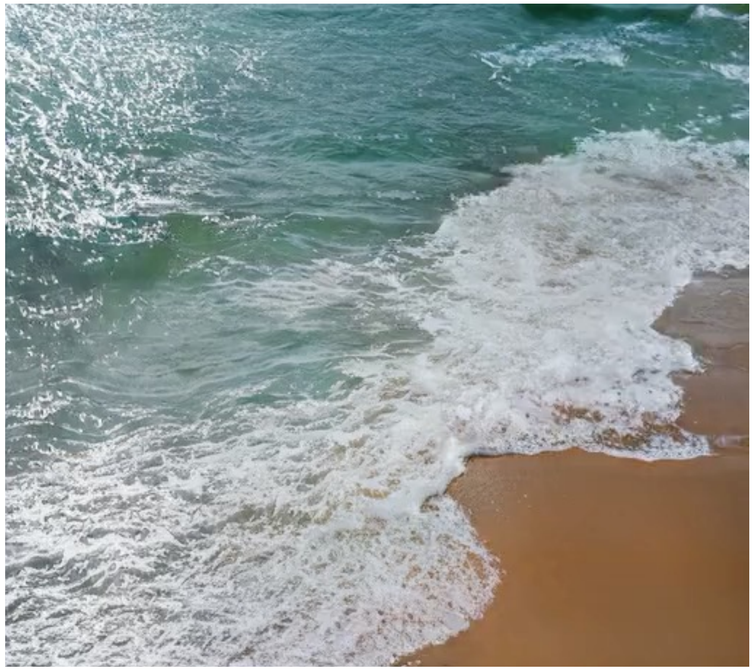


Sample

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Application (1): Content Generation

StyleGAN3 example images



Karras et al. Alias-Free Generative Adversarial Networks, NeurIPS 2021
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Application (2): Representation Learning

Learning from limited labels

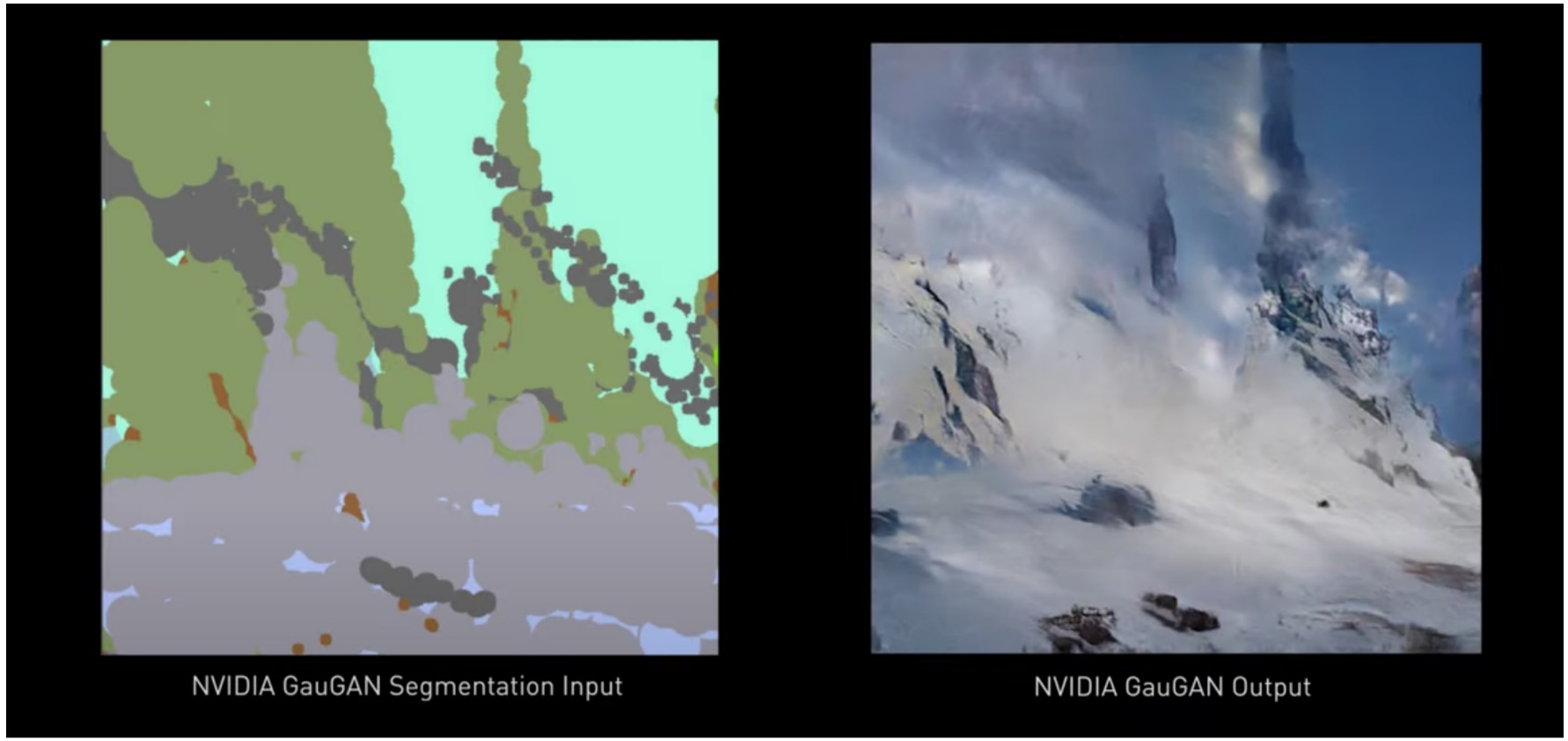
Zhang et al., DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort, CVPR 2021

Li et al., Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization, CVPR 2021

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Application (3): Artistic Tools

NVIDIA GauGAN



Park et al., Semantic Image Synthesis with Spatially-Adaptive Normalization, CVPR 2019

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

2022 / 2023 : The year of generative modeling?

Parti

Pathways Autoregressive Text-to-Image Model

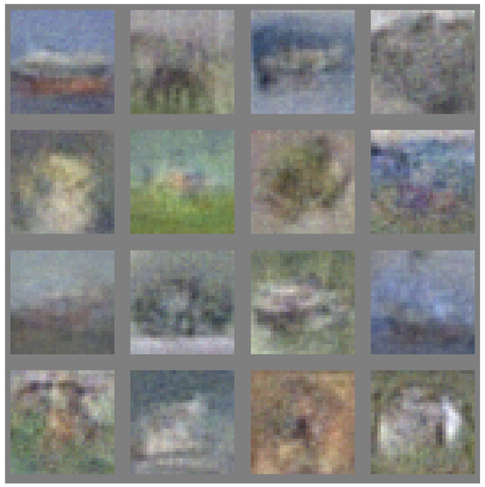
DALL·E 2

Stable Diffusion

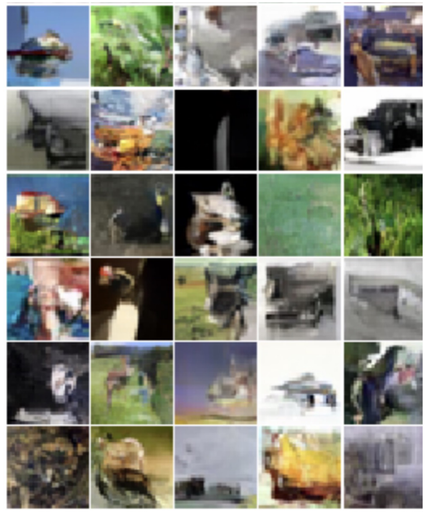
Where we came from

VAEs, 2013

GANs, 2014

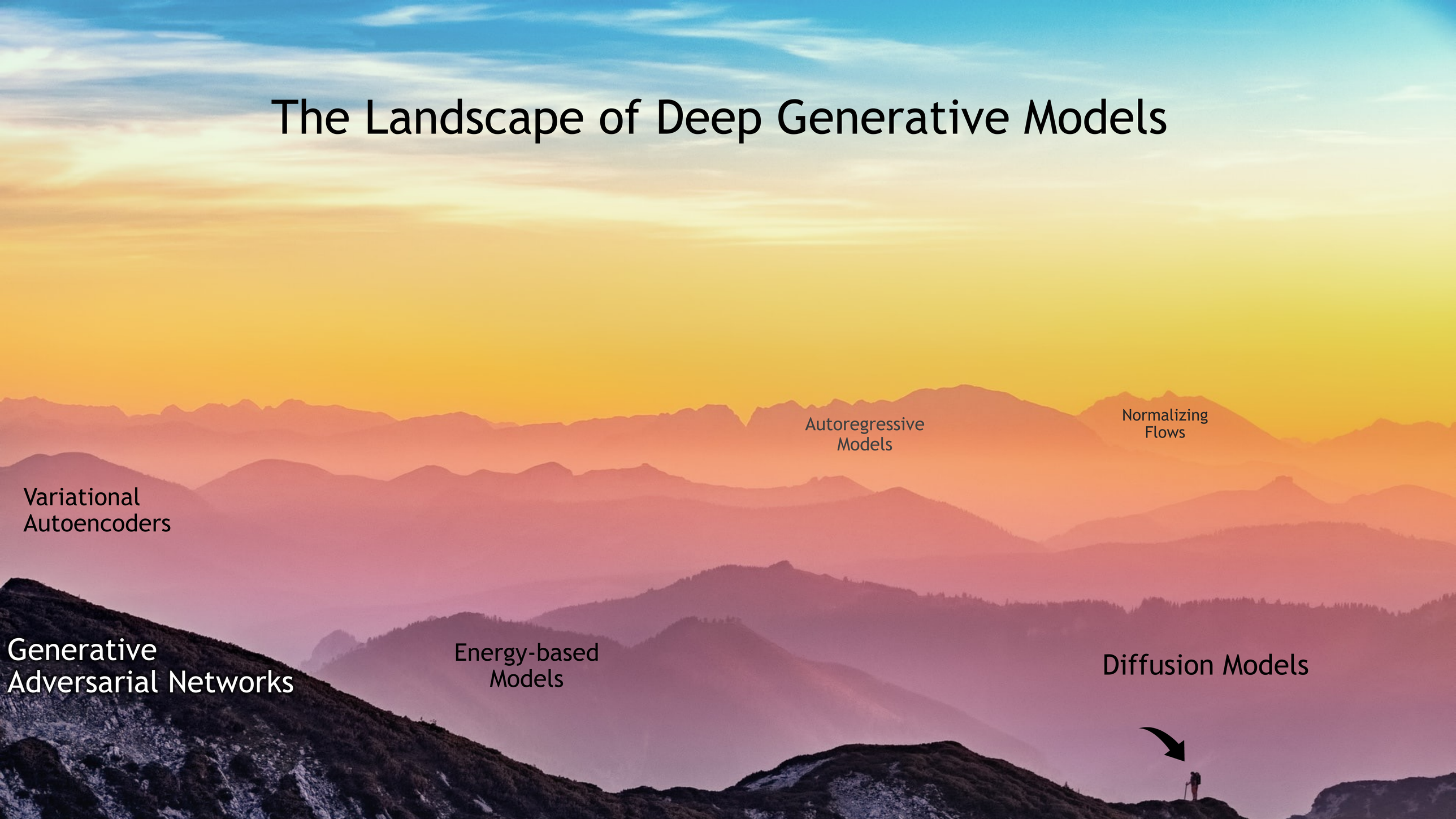


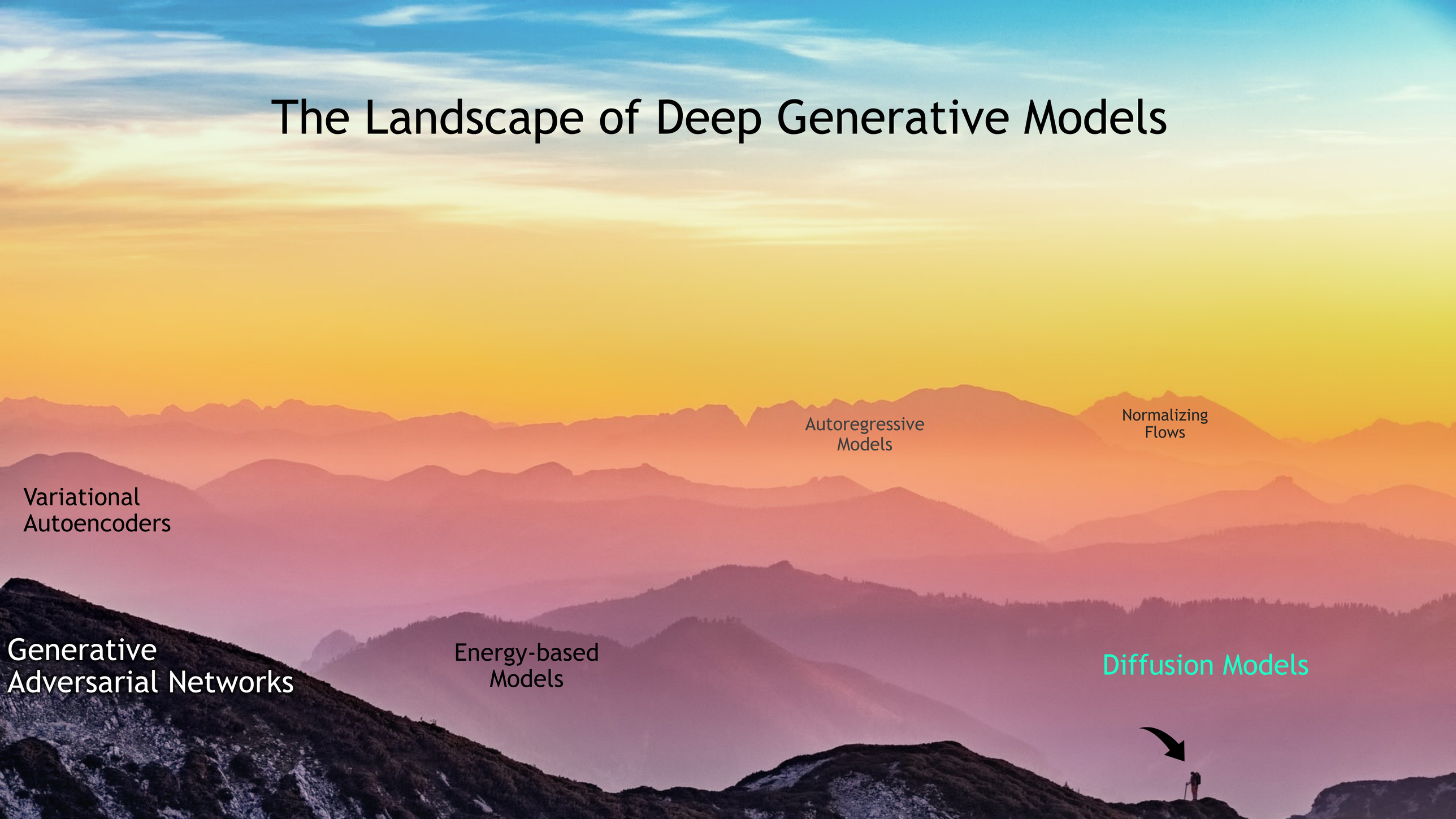
PixelCNN, 2016



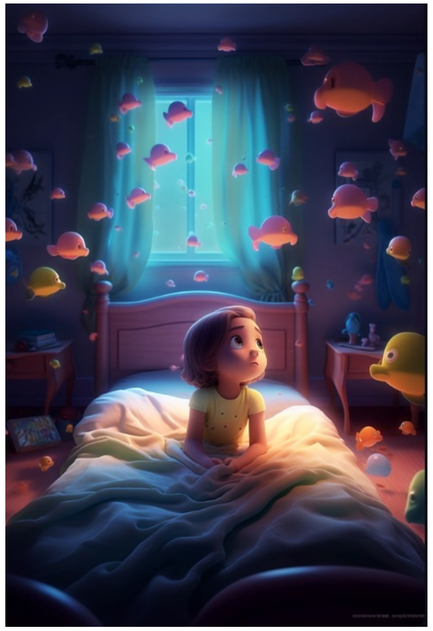
BigGAN, 2019

Imagen, 2022


The Landscape of Deep Generative Models
Autoregressive Models
Normalizing Flows
Variational Autoencoders
Generative Adversarial Networks
Energy-based Models
Diffusion Models

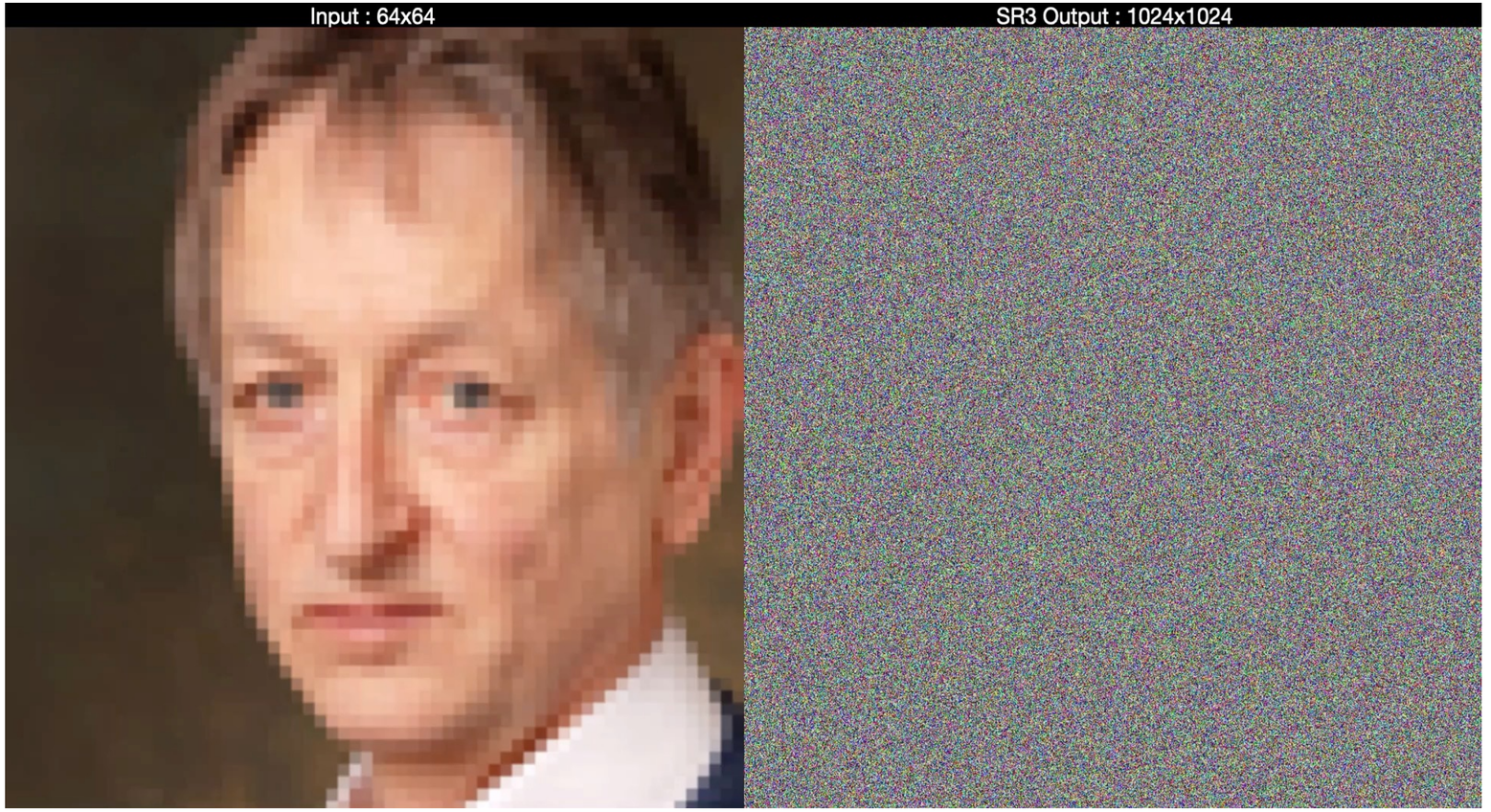

The Landscape of Deep Generative Models
Autoregressive Models
Normalizing Flows
Variational Autoencoders
Generative Adversarial Networks
Energy-based Models
Diffusion Models

AI Art



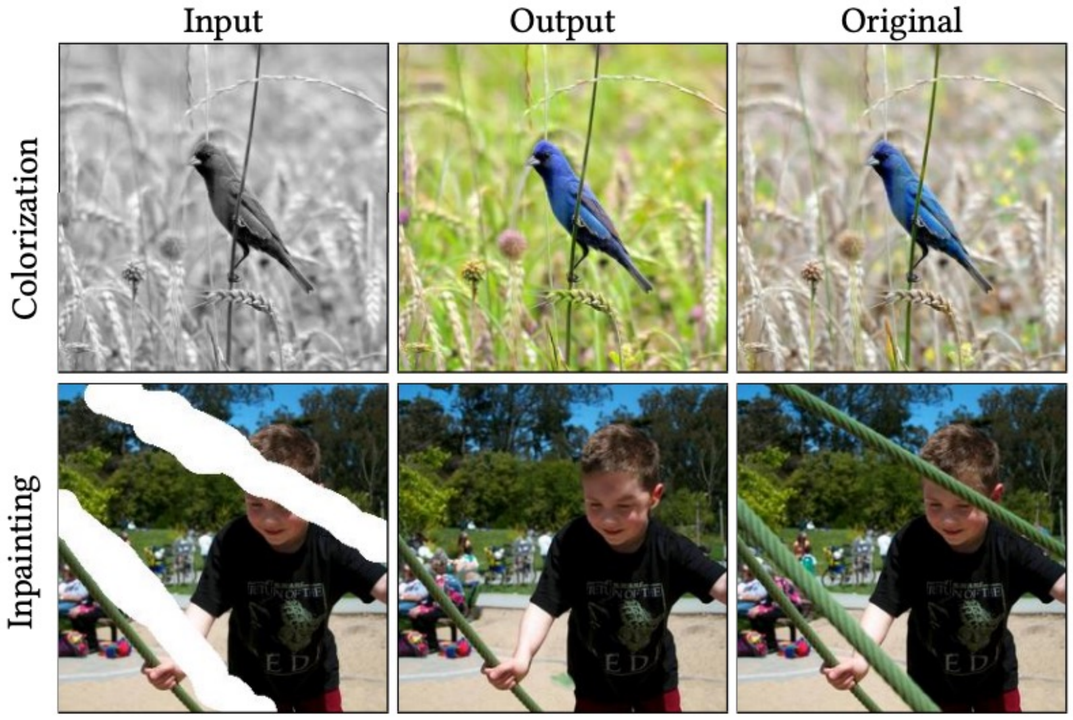
Midjourney v5

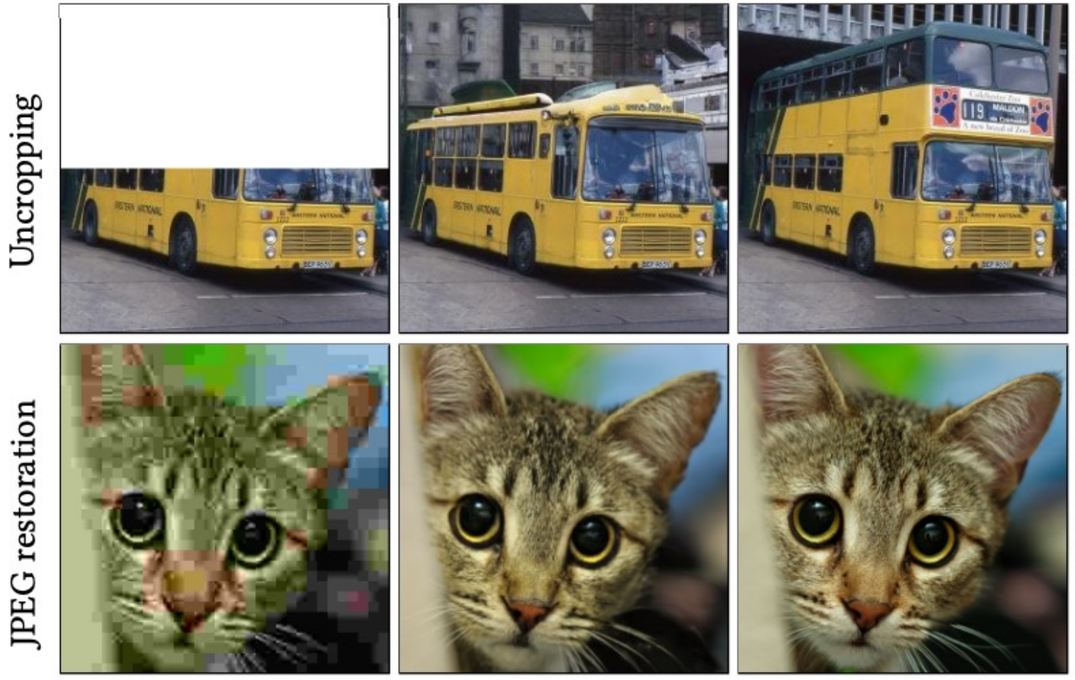
Applications: Super-resolution



Palette: Image-to-Image Diffusion Models, 2022

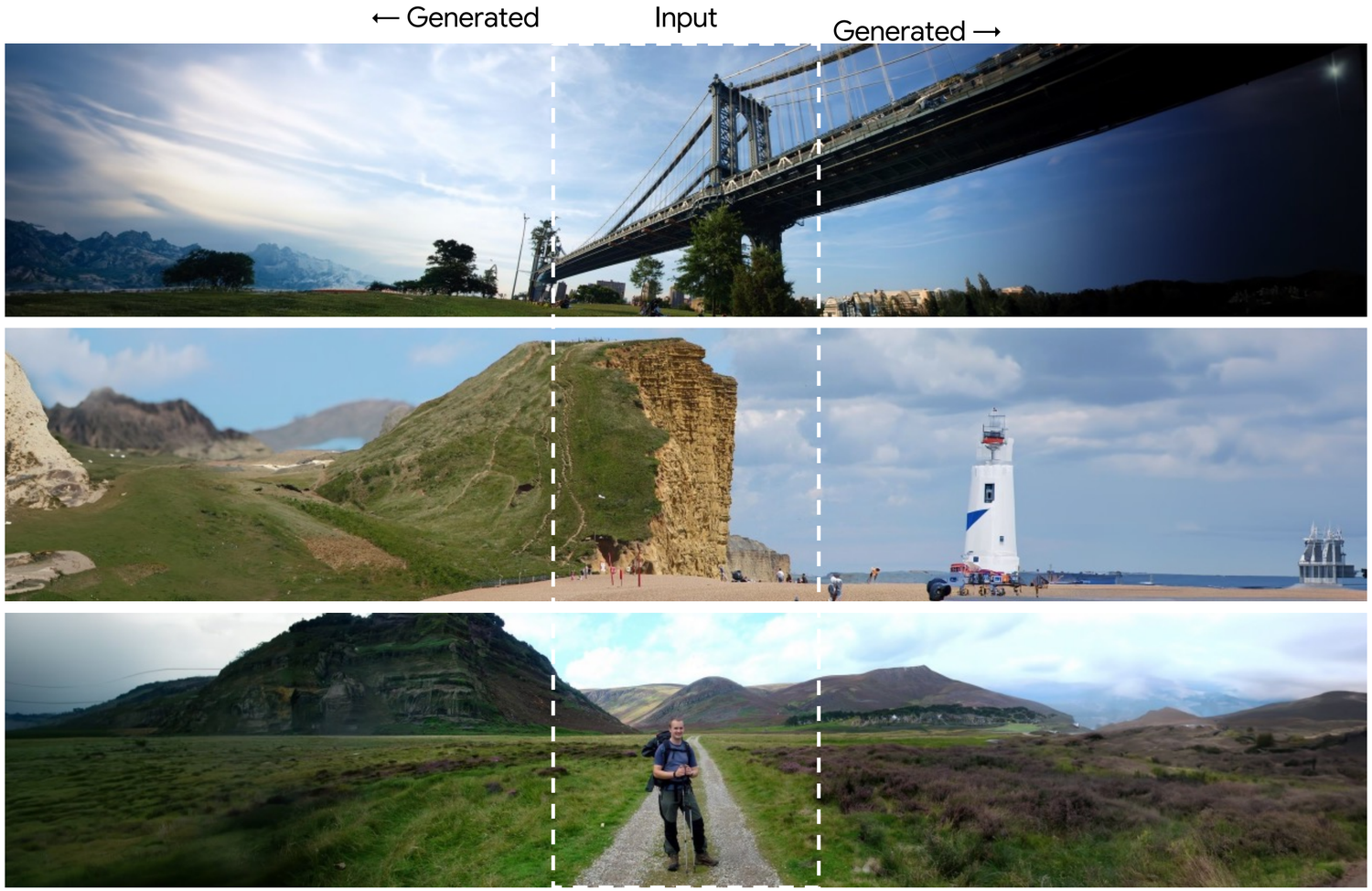
Applications: Colorization, Inpainting, Restoration





Palette: Image-to-Image Diffusion Models, 2022

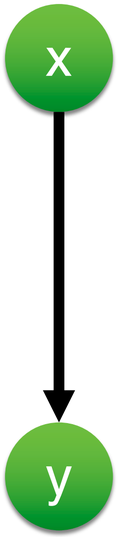
Applications: Outfilling



Palette: Image-to-Image Diffusion Models, 2022

Variational Autoencoders

Classifying chocolate



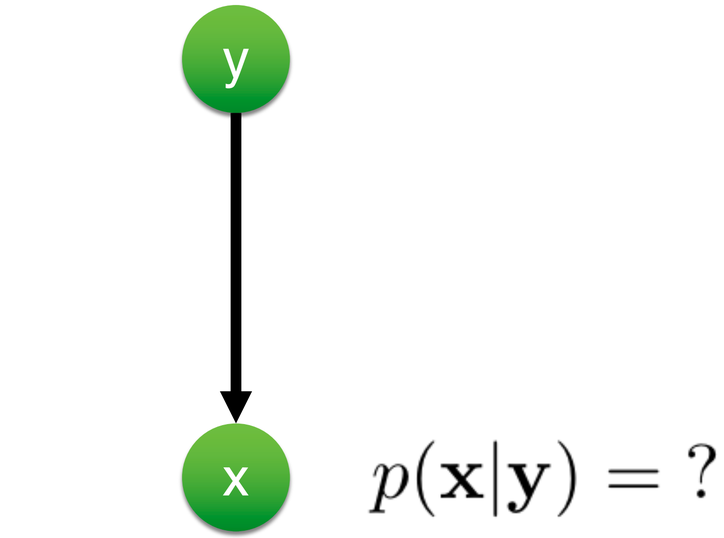
$$p(\mathbf{y}|\mathbf{x}) = \text{Categorical}(\mathbf{y}; \mathbf{f}(\mathbf{x}))$$

“chocolate”

slide by Durk Kingma

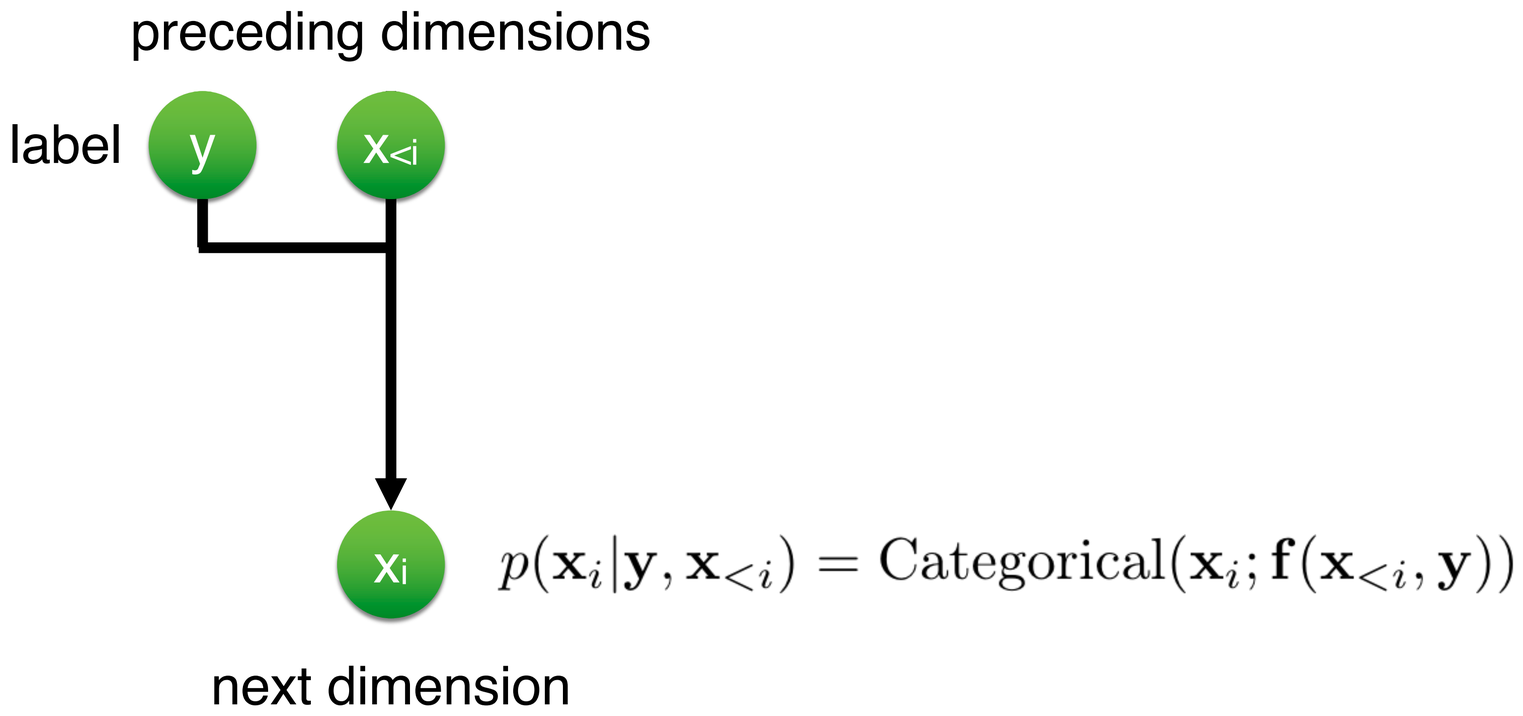
Generating chocolate

“chocolate”



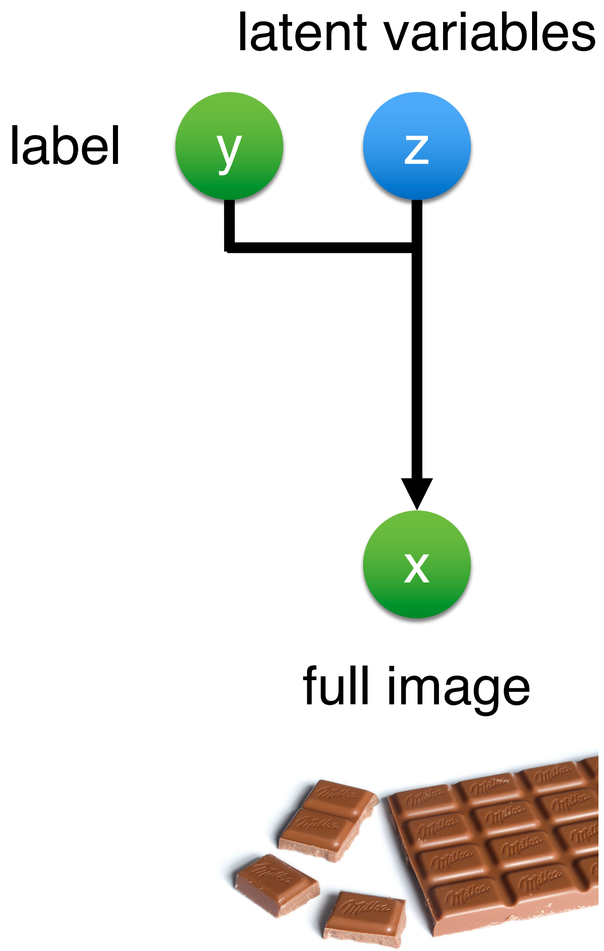
slide by Durk Kingma

Autoregressive approach



slide by Durk Kingma

Latent variable approach

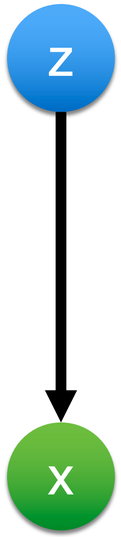


$$
\begin{aligned}
p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) \\
p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}, \mathbf{y}), \sigma^2 \mathbf{I}) \\
p_\theta(\mathbf{x}|\mathbf{y}) &= \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{y}, \mathbf{z}) p(\mathbf{z})\, d\mathbf{z} \\
&= \text{flexible distribution}
\end{aligned}
$$

slide by Durk Kingma

Latent variable approach - without y

latent variables



full image

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \sigma^2\mathbf{I})$$

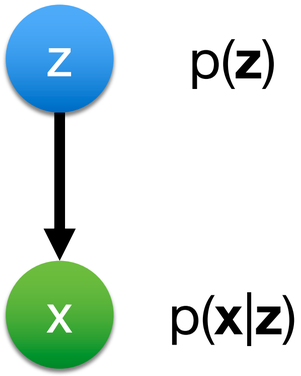
$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})\, d\mathbf{z}$$

$$= \text{flexible distribution}$$

slide by Durk Kingma

Optimization

$$
\begin{aligned}
p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) \\
p_\theta(\mathbf{x}|\mathbf{z}) &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \sigma^2 \mathbf{I}) \\
p_\theta(\mathbf{x}) &= \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})\, d\mathbf{z} \\
&= \text{flexible distribution}
\end{aligned}
$$



Generative model
p(x,z)

- Problem:
  - Marginal likelihood p(x) is intractable
  - So can’t do maximum likelihood directly

slide by Durk Kingma

Variational Autoencoders (VAEs)

- We introduce an **inference model** q(z|x)

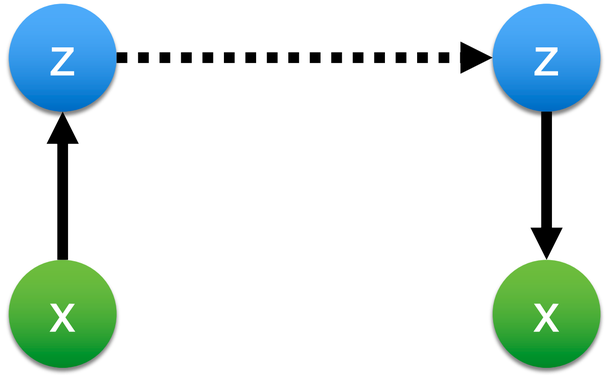
$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\Sigma}_\phi(\mathbf{x}))$$

- This allows us to efficiently optimize the log-likelihood, through the **evidence lower bound** (ELBO).

$$\log p_{\theta,\phi}(\mathbf{x}) \geq \mathrm{ELBO}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]$$

- We optimize q(z|x) and p(x,z) jointly w.r.t. ELBO

- Bound is tight with the right q(z|x)



Inference model
q(z|x)

Generative model
p(x,z)

slide by Durk Kingma

[Kingma and Welling, 2013]

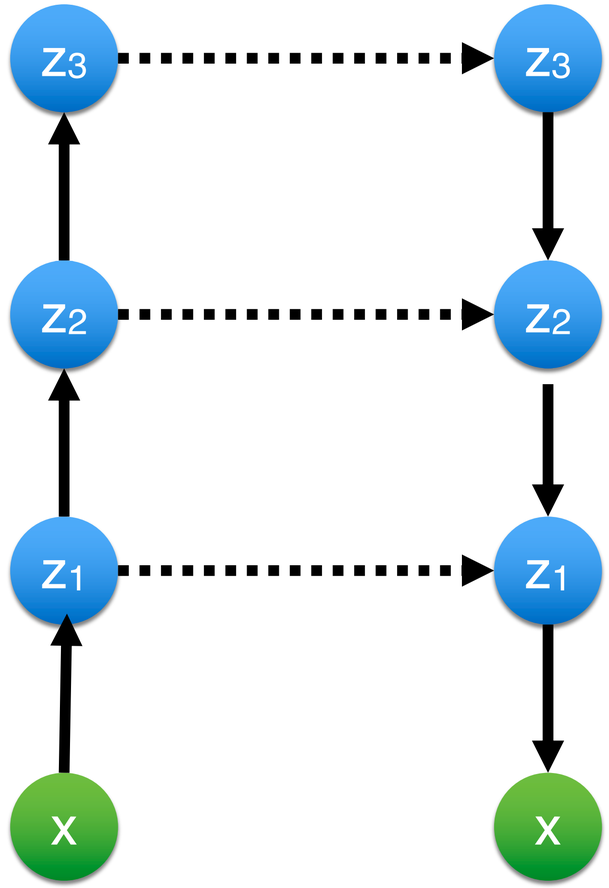
Hierarchical VAEs

- “Flat” VAEs suffer from simple priors
- Making both inference model and generative model hierarchical

$$q_\phi(\mathbf{z}_{1,2,3}|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1)q_\phi(\mathbf{z}_3|\mathbf{z}_2)$$

$$p_\theta(\mathbf{z}_{1,2,3}) = p_\theta(\mathbf{z}_3)p_\theta(\mathbf{z}_2|\mathbf{z}_3)p_\theta(\mathbf{z}_1|\mathbf{z}_2)p_\theta(\mathbf{x}|\mathbf{z}_1)$$

- Better likelihoods are achieved with hierarchies of latent variables



Inference model
q(z|x)

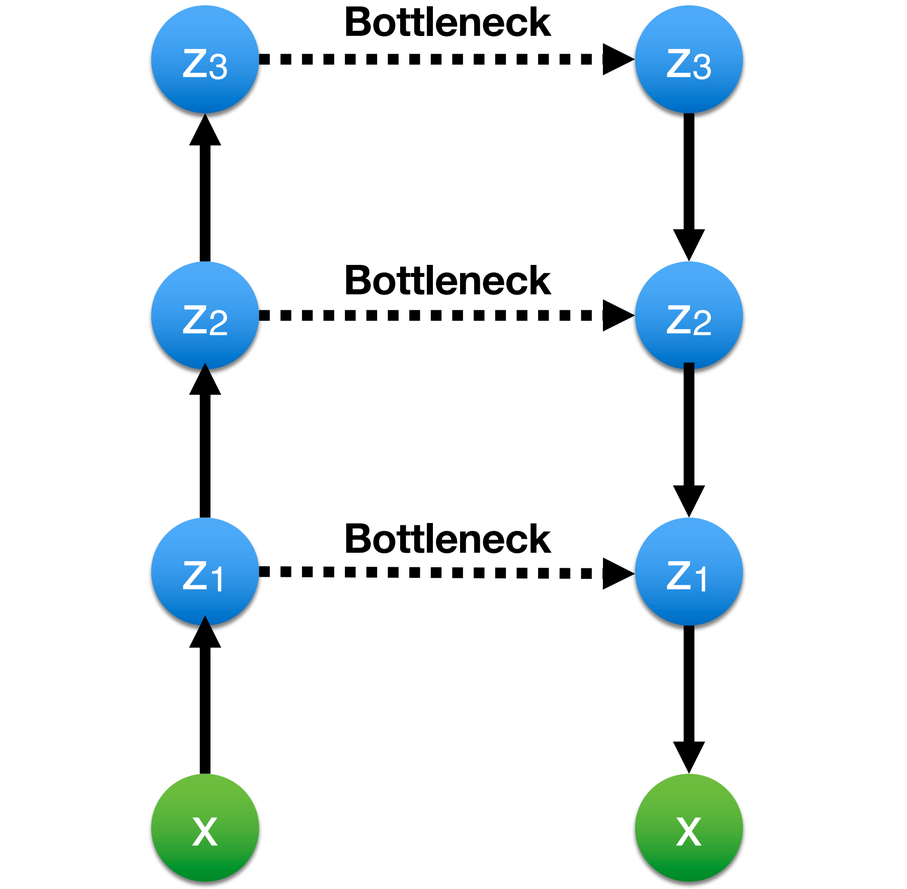
Generative model
p(x,z)

slide by Durk Kingma

[Kingma and Welling, 2017]

VAEs: challenges

- Optimization can be difficult for large models
- The ELBO enforces an **information bottleneck** (through its loss function) at the latent variables 'z', making VAE optimization prone to **bad local minima**.
- **Posterior collapse** is a dreaded bad local minimum where the latents do not transmit any information.



slide by Durk Kingma

[Kingma and Welling, 2013]

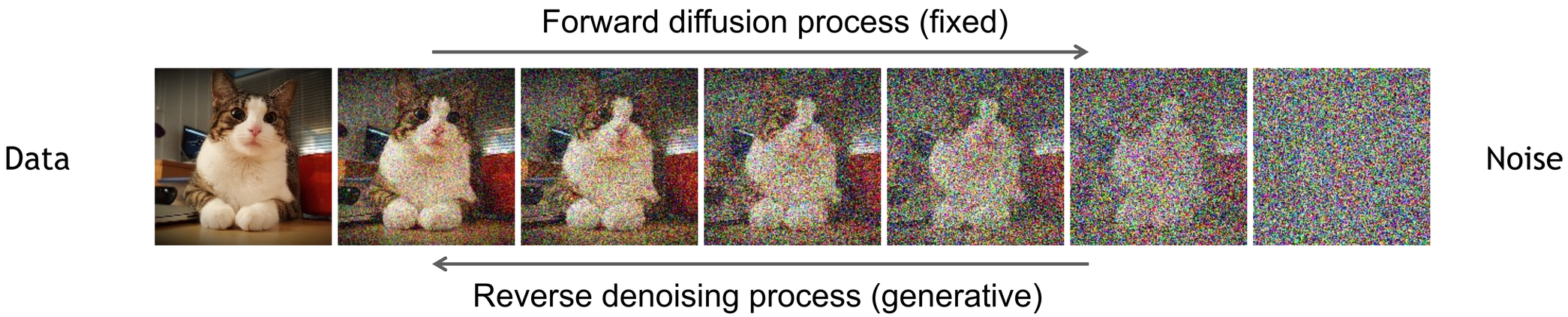
Diffusion Models

Denoising Diffusion Models

Learning to generate by denoising

Denoising diffusion models consist of two processes:

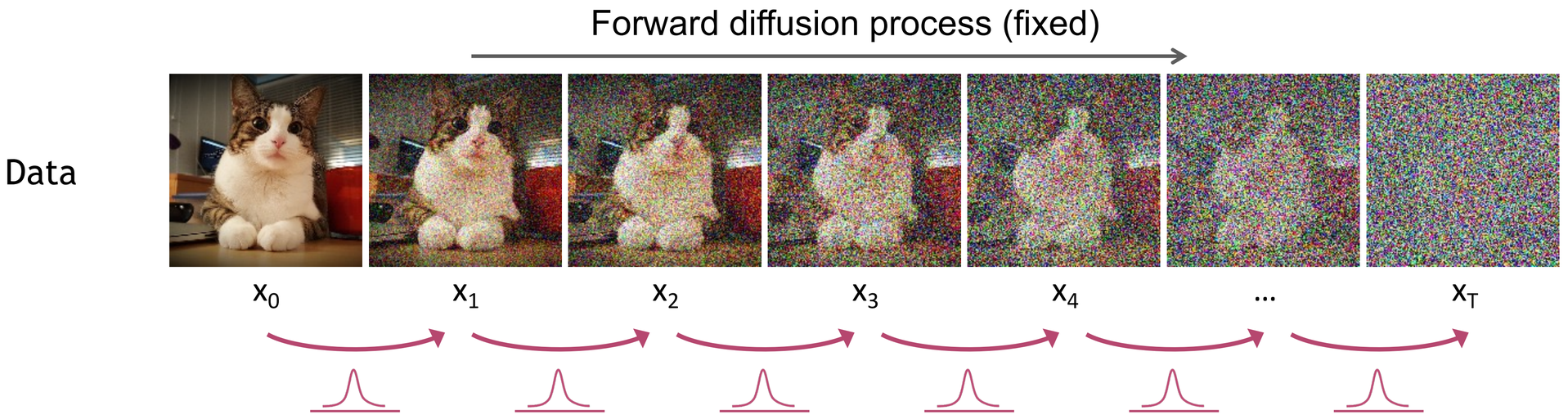
- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Forward Diffusion Process

The formal definition of the forward process in T steps:



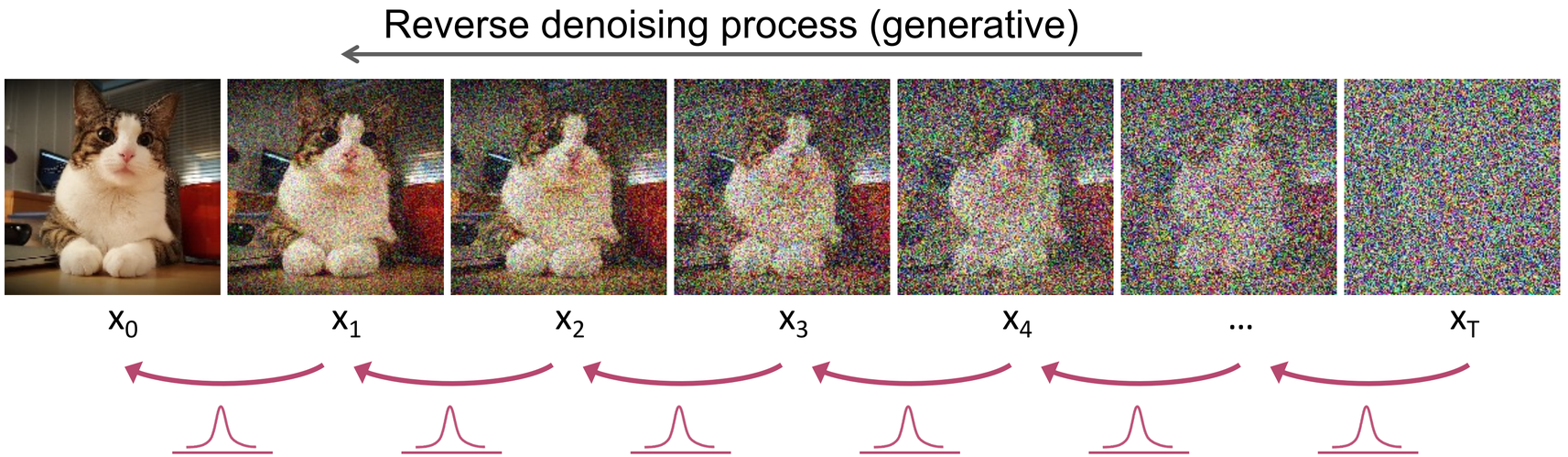
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

Similar to the inference model in hierarchical VAEs.

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Reverse Denoising Process

Formal definition of forward and reverse processes in T steps:



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}, \sigma_t^2\mathbf{I})$$

Trainable network
(U-net, Denoising Autoencoder)

➡

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Similar to the generative model in hierarchical VAEs.

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Slides made from:

Understanding Diffusion Models: A Unified Perspective

An intuitive, accessible tutorial on diffusion models.

AUTHORS

Calvin Luo

AFFILIATIONS

Google Research [3]

Brown University

PUBLISHED

Aug. 26, 2022

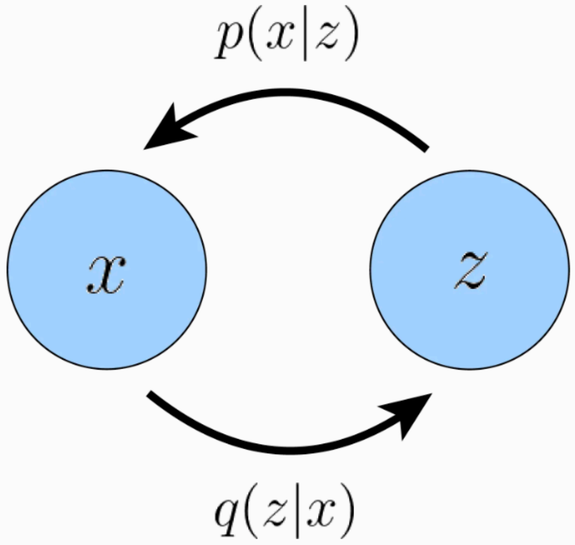
https://www.calvinyluo.com/2022/08/26/diffusion-tutorial.html

Variational autoencoders

$$\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}\mid\boldsymbol{x})}\right] = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}\mid\boldsymbol{z})p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}\mid\boldsymbol{x})}\right] \tag{17}$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\right] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}\mid\boldsymbol{x})}\right] \tag{18}$$

$$= \underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}\mid\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{z}\mid\boldsymbol{x})\,||\,p(\boldsymbol{z}))}_{\text{prior matching term}} \tag{19}$$



Variational autoencoders

A defining feature of the VAE is how the ELBO is optimized jointly over parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. The encoder of the VAE is commonly chosen to model a multivariate Gaussian with diagonal covariance, and the prior is often selected to be a standard multivariate Gaussian:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x})\mathbf{I}) \tag{20}$$

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \mathbf{I}) \tag{21}$$

Variational autoencoders

Then, the KL divergence term of the ELBO can be computed analytically, and the reconstruction term can be approximated using a Monte Carlo estimate. Our objective can then be rewritten

$$\begin{aligned}
&\arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \\
&\approx \arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}^{(l)}) - \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \qquad (22)
\end{aligned}$$

Variational autoencoders

Then, the KL divergence term of the ELBO can be computed analytically, and the reconstruction term can be approximated using a Monte Carlo estimate. Our objective can then be rewritten

$$\begin{aligned}
&\arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \\
&\approx \arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}^{(l)}) - \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \qquad (22)
\end{aligned}$$

For example, samples from a normal distribution $x \sim \mathcal{N}(x; \mu, \sigma^2)$ with arbitrary mean μ and variance σ^2 can be rewritten as:

$$x = \mu + \sigma\epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; 0, \mathrm{I})$$

Variational autoencoders

Then, the KL divergence term of the ELBO can be computed analytically, and the reconstruction term can be approximated using a Monte Carlo estimate. Our objective can then be rewritten

$$\begin{aligned}
&\arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \\
&\approx \arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}^{(l)}) - \mathcal{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p(\boldsymbol{z})) \qquad (22)
\end{aligned}$$

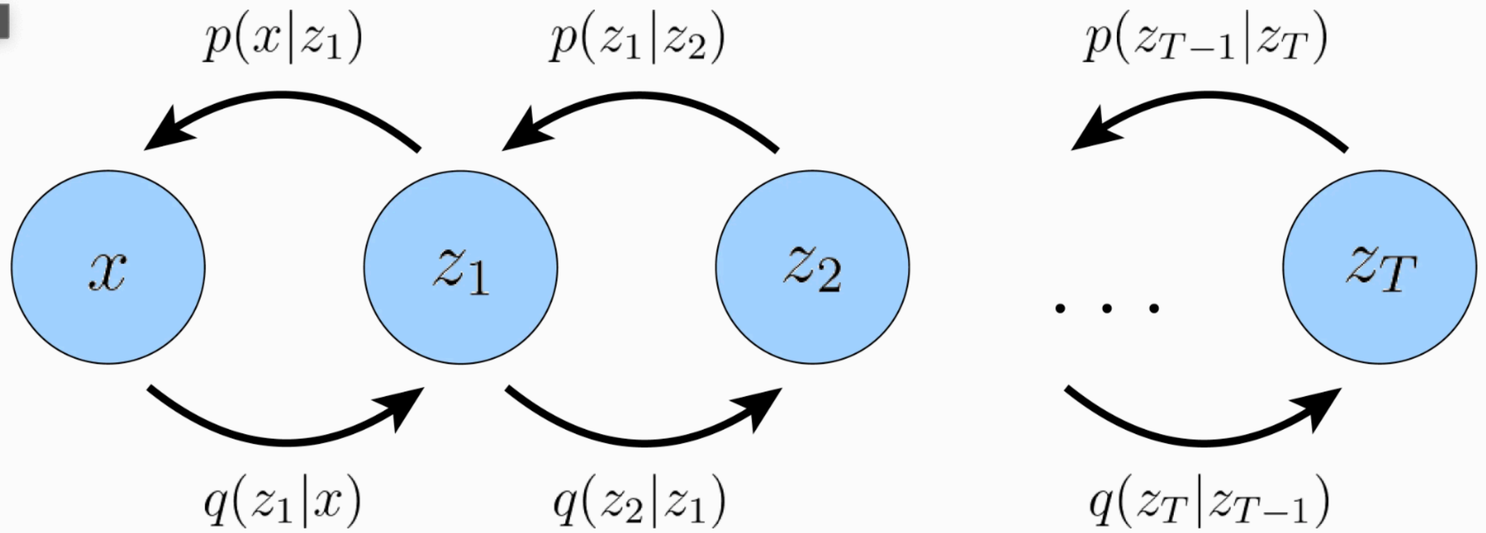
For example, samples from a normal distribution $x \sim \mathcal{N}(x; \mu, \sigma^2)$ with arbitrary mean μ and variance σ^2 can be rewritten as:

$$x = \mu + \sigma\epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; 0, \mathrm{I})$$

In a VAE, each $\boldsymbol{z}$ is thus computed as a deterministic function of input $\boldsymbol{x}$ and auxiliary noise variable $\boldsymbol{\epsilon}$:

$$\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}) \odot \boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$$

Hierarchical VAEs



A Markovian Hierarchical Variational Autoencoder. The generative process is modeled as a Markov chain, where each latent is generated only from the previous latent.

$$p(\boldsymbol{x}, \boldsymbol{z}_{1:T}) = p(\boldsymbol{z}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1} \mid \boldsymbol{z}_t) \tag{23}$$

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{t=2}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}) \tag{24}$$

Hierarchical VAEs

Then, we can easily extend the ELBO to be:

$$\log p(\boldsymbol{x}) = \log \int p(\boldsymbol{x}, \boldsymbol{z}_{1:T}) d\boldsymbol{z}_{1:T} \tag{25}$$

$$= \log \int \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T}) q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x})} d\boldsymbol{z}_{1:T} \tag{26}$$

$$= \log \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})} \left[\frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x})} \right] \tag{27}$$

$$\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x})} \right] \tag{28}$$

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}_{1:T})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T} \mid \boldsymbol{x})} \right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{1:T}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{z}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{z}_1) \prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{z}_{t-1} \mid \boldsymbol{z}_t)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{t=2}^{T} q_{\boldsymbol{\phi}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})} \right] \tag{29}$$

Variational Diffusion Models

The easiest way to think of a Variational Diffusion Model (VDM) [1, 2, 10] is simply as a Markov Hierarchical Variational Autoencoder with three key restrictions:

$$q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}) = q_\phi(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{t=2}^{T} q_\phi(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$$

Variational Diffusion Models

The easiest way to think of a Variational Diffusion Model (VDM) [1, 2, 10] is simply as a Markov Hierarchical Variational Autoencoder with three key restrictions:

- The latent dimension is exactly equal to the data dimension

$$q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}) = q_\phi(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{t=2}^{T} q_\phi(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$$

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

Variational Diffusion Models

The easiest way to think of a Variational Diffusion Model (VDM) [1, 2, 10] is simply as a Markov Hierarchical Variational Autoencoder with three key restrictions:

- The latent dimension is exactly equal to the data dimension
- The structure of the latent encoder at each timestep is not learned; it is pre-defined as a linear Gaussian model. In other words, it is a Gaussian distribution centered around the output of the previous timestep

$$q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}) = q_\phi(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{t=2}^{T} q_\phi(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$$

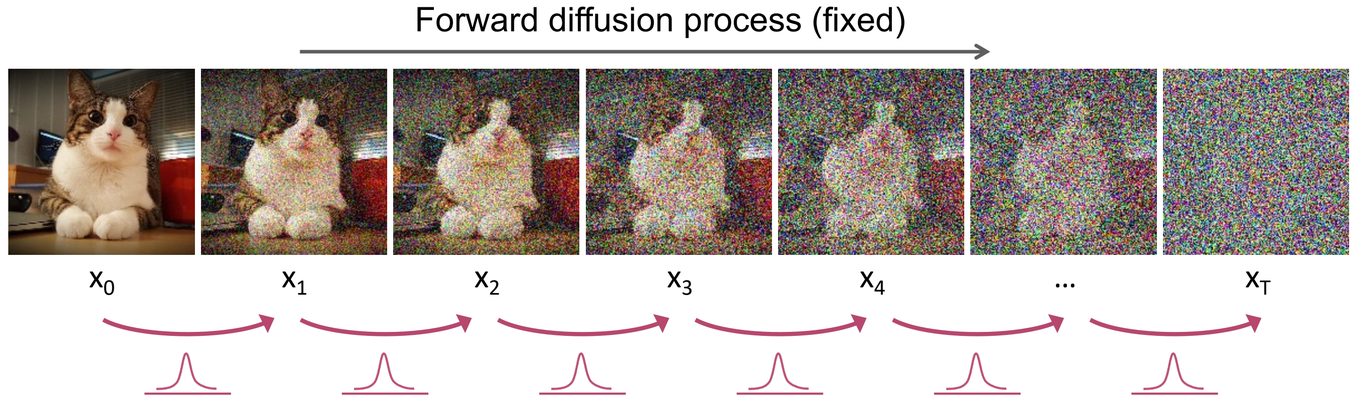
$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

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

Variational Diffusion Models

The easiest way to think of a Variational Diffusion Model (VDM) [1, 2, 10] is simply as a Markov Hierarchical Variational Autoencoder with three key restrictions:

- The latent dimension is exactly equal to the data dimension
- The structure of the latent encoder at each timestep is not learned; it is pre-defined as a linear Gaussian model. In other words, it is a Gaussian distribution centered around the output of the previous timestep
- The Gaussian parameters of the latent encoders vary over time in such a way that the distribution of the latent at final timestep T is a standard Gaussian

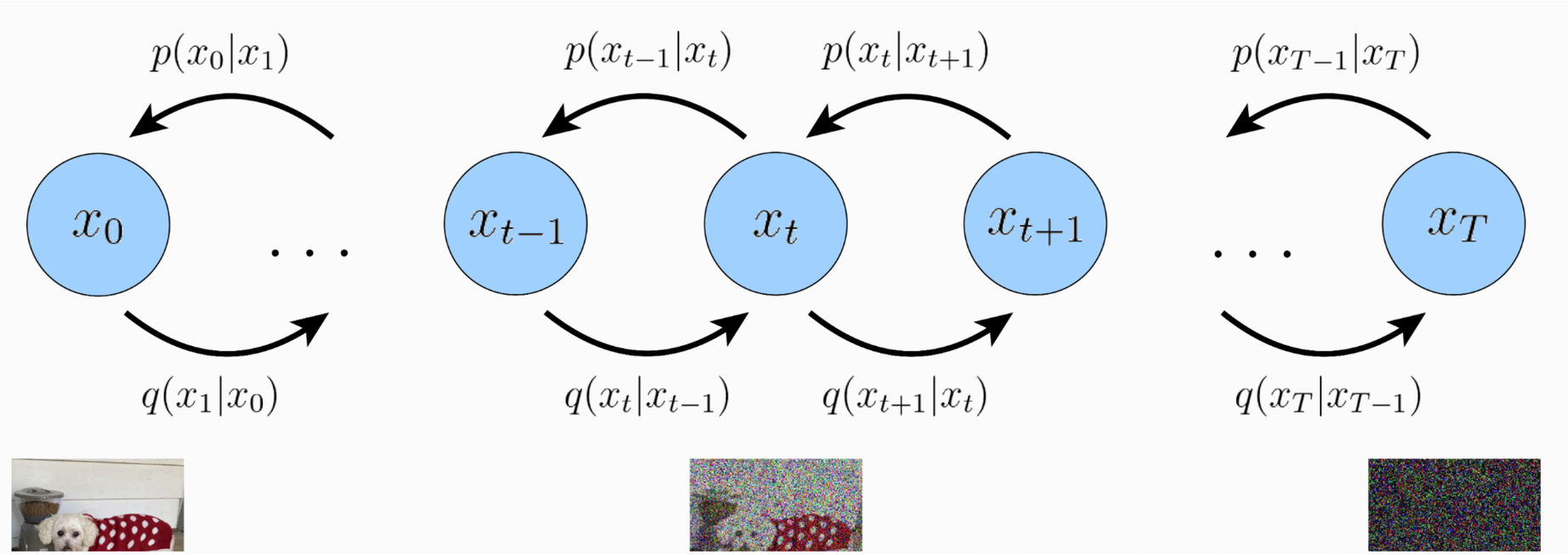


$$q_\phi(\boldsymbol{z}_{1:T} \mid \boldsymbol{x}) = q_\phi(\boldsymbol{z}_1 \mid \boldsymbol{x}) \prod_{t=2}^{T} q_\phi(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1})$$

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

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

Variational Diffusion Models



Note that our encoder distributions $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ are no longer parameterized by $\boldsymbol{\phi}$, as they are completely modeled as Gaussians with defined mean and variance parameters at each timestep

$$p(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

$$p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \mathbf{I})$$

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

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

Variational Diffusion Models

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\prod_{t=1}^{T-1} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_T \mid \boldsymbol{x}_{T-1})\prod_{t=1}^{T-1} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T \mid \boldsymbol{x}_{T-1})}\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\sum_{t=1}^{T-1} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}\right]$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{q(\boldsymbol{x}_{T-1}|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_{T-1}) \,||\, p(\boldsymbol{x}_T))\right]}_{\text{prior matching term}} - \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_{t-1},\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) \,||\, p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+1}))\right]}_{\text{consistency term}}$$

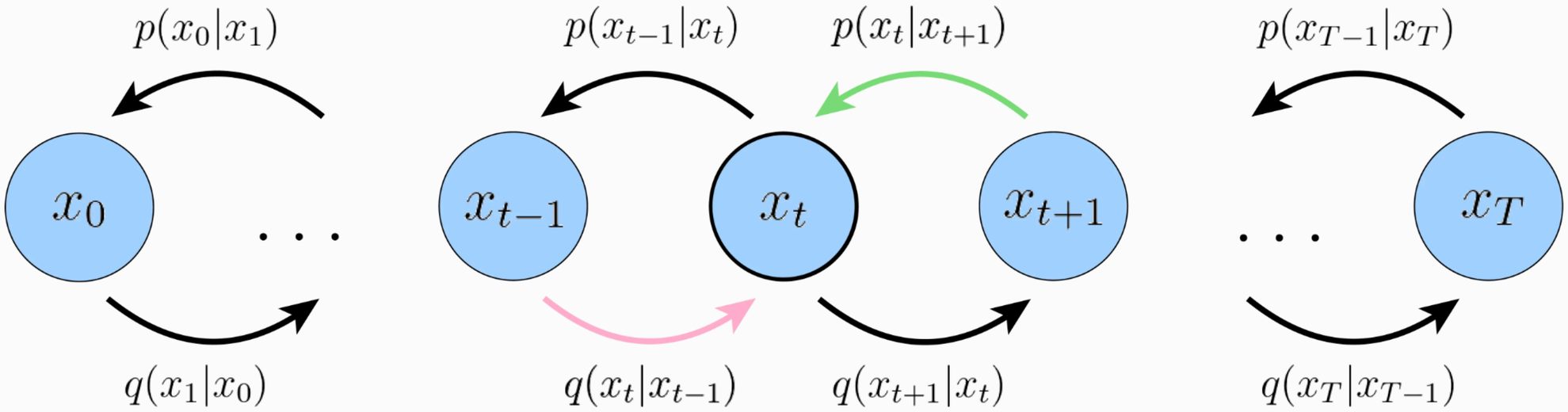
Variational Diffusion Models

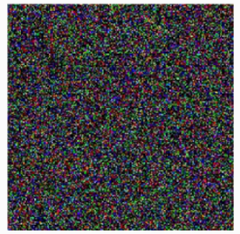
1. $\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]$ can be interpreted as a *reconstruction term*, predicting the log probability of the original data sample given the first-step latent. This term also appears in a vanilla VAE, and can be trained similarly.

2. $\mathbb{E}_{q(\boldsymbol{x}_{T-1}|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_{T-1}) \mid\mid p(\boldsymbol{x}_T))\right]$ is a *prior matching term*; it is minimized when the final latent distribution matches the Gaussian prior. This term requires no optimization, as it has no trainable parameters; furthermore, as we have assumed a large enough T such that the final distribution is Gaussian, this term effectively becomes zero.

3. $\mathbb{E}_{q(\boldsymbol{x}_{t-1},\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) \mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+1}))\right]$ is a *consistency term*; it endeavors to make the distribution at $\boldsymbol{x}_t$ consistent, from both forward and backward processes. That is, a denoising step from a noisier image should match the corresponding noising step from a cleaner image, for every intermediate timestep; this is reflected mathematically by the KL Divergence. This term is minimized when we train $p_\theta(\boldsymbol{x}_t \mid \boldsymbol{x}_{t+1})$ to match the Gaussian distribution $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$, which is defined in Equation $\underline{31}$.

Variational Diffusion Models





A VDM can be optimized by ensuring that for every intermediate latent, the posterior from the latent above it matches the Gaussian corruption of the latent before it. In this figure, for each intermediate latent, we minimize the difference between the distributions represented by the pink and green arrows.

Variational Diffusion Models

$$\mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right] = \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)}{q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\,\cancel{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}}{\cancel{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}}\right]$$

Variational Diffusion Models

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T) p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)}{q(\boldsymbol{x}_T \mid \boldsymbol{x}_0)} + \sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)} \right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} [\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)] + \mathbb{E}_{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} \left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T \mid \boldsymbol{x}_0)} \right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|\boldsymbol{x}_0)} \left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)} \right]$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} [\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,||\, p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} [\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

Variational Diffusion Models

The ELBO can therefore be decomposed as a sum of individual terms that are expectations of at most one random variable at a time. This formulation also has an elegant interpretation, which is revealed when inspecting each term individually:

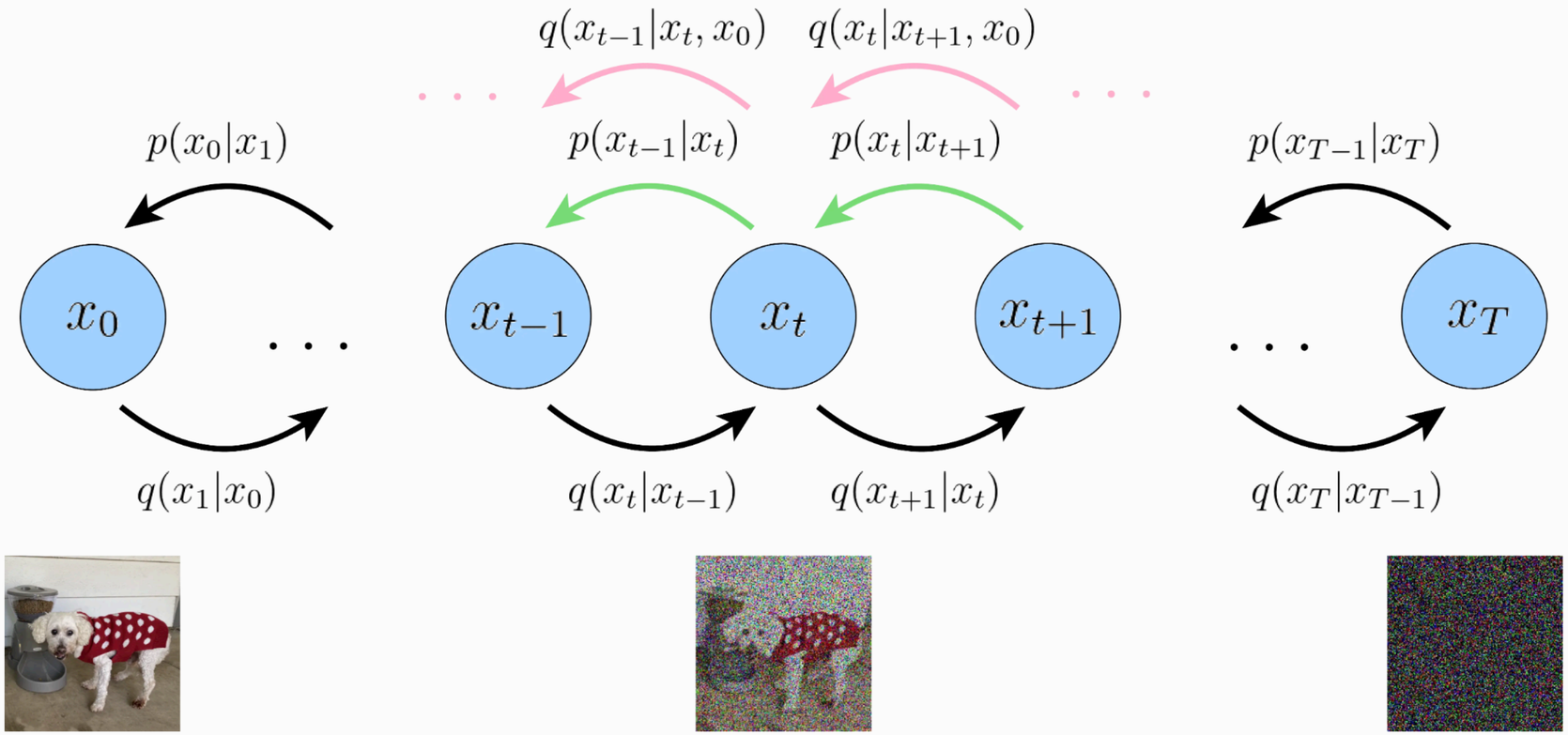
1. $\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]$ can be interpreted as a *reconstruction term*; like its analogue in the ELBO of a vanilla VAE, this term can be approximated and optimized using a Monte Carlo estimate.

2. $\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \mid\mid p(\boldsymbol{x}_T))$ is a *prior matching term*; it represents how close the distribution of the final noisified input is to the standard Gaussian prior. It has no trainable parameters, and is also equal to zero under our assumptions.

3. $\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \mid\mid p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))\right]$ is a *denoising matching term*. We learn desired denoising transition step $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ as an approximation to tractable, ground-truth denoising transition step $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$. The $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ transition step can act as a ground-truth signal, since it defines how to denoise a noisy image $\boldsymbol{x}_t$ with access to what the final, completely denoised image $\boldsymbol{x}_0$ should be. This term is therefore minimized when the two denoising steps match as closely as possible, as measured by their KL Divergence.

Variational Diffusion Models

A visual interpretation of this ELBO decomposition is depicted in the figure below:



A VDM can also be optimized by learning the denoising step for each individual latent by matching it with a tractably computed ground-truth denoising step. This is once again denoted visually by matching the distributions represented by the green arrows with those of the pink arrows. Artistic liberty is at play here; in the full picture, each pink arrow must also stem from the ground-truth image, as it is also a conditioning term.

Variational Diffusion Models

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)}$$

Variational Diffusion Models

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

$$\begin{aligned} q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)} \\ &= \frac{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I}) \mathcal{N}(\boldsymbol{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})} \end{aligned}$$

Variational Diffusion Models

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

$$
\begin{aligned}
q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)} \\
&= \frac{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I}) \mathcal{N}(\boldsymbol{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})} \\
&\propto \mathcal{N}(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}_{\mu_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\boldsymbol{\Sigma}_q(t)})
\end{aligned}
$$

$\boldsymbol{\Sigma}_q(t) = \sigma_q^2(t)\mathbf{I}$, where:

$$\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$$

Variational Diffusion Models

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

$$
\begin{aligned}
q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)} \\
&= \frac{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I}) \mathcal{N}(\boldsymbol{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})} \\
&\propto \mathcal{N}(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}_{\mu_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{I}}_{\boldsymbol{\Sigma}_q(t)})
\end{aligned}
$$

$\boldsymbol{\Sigma}_q(t) = \sigma_q^2(t)\mathbf{I}$, where:

$$\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$$

Variational Diffusion Models

$$\underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,||\, p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

In order to match approximate denoising transition step $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ to ground-truth denoising transition step $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ as closely as possible, we can also model it as a Gaussian. Furthermore, as all α terms are known to be frozen at each timestep, we can immediately construct the variance of the approximate denoising transition step to also be $\boldsymbol{\Sigma}_q(t) = \sigma_q^2(t)\mathbf{I}$. We must parameterize its mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as a function of $\boldsymbol{x}_t$, however, since $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ does not condition on $\boldsymbol{x}_0$.

Variational Diffusion Models

$$\underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,||\, p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

In order to match approximate denoising transition step $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ to ground-truth denoising transition step $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ as closely as possible, we can also model it as a Gaussian. Furthermore, as all α terms are known to be frozen at each timestep, we can immediately construct the variance of the approximate denoising transition step to also be $\boldsymbol{\Sigma}_q(t) = \sigma_q^2(t)\mathbf{I}$. We must parameterize its mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as a function of $\boldsymbol{x}_t$, however, since $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ does not condition on $\boldsymbol{x}_0$.

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)) \\
&= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right)
\end{aligned}$$

Variational Diffusion Models

$$\underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,||\, p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

In orde
denois
Gaussi
immed
$\boldsymbol{\Sigma}_q(t)$
$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ does not condition on $\boldsymbol{x}_0$.

We utilize the fact that the KL Divergence between two Gaussian distributions is:

$$\mathcal{D}_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \,||\, \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)) = \frac{1}{2}\left[\log \frac{|\,\boldsymbol{\Sigma}_y\,|}{|\,\boldsymbol{\Sigma}_x\,|} - d + \mathrm{tr}(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_x) + (\boldsymbol{\mu}_y - \boldsymbol{\mu}_x)^T \boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\mu}_y - \boldsymbol{\mu}_x)\right]$$

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)) \\
&= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\|_2^2\right]
\end{aligned}$$

Variational Diffusion Models

$$= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right]$$

Variational Diffusion Models

$$= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right]$$

where we have written $\boldsymbol{\mu}_q$ as shorthand for $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ as shorthand for $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ for brevity. In other words, we want to optimize a $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ that matches $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, which from our derived Equation $\underline{53}$, takes the form:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{58}$$

Variational Diffusion Models

$$= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right]$$

where we have written $\boldsymbol{\mu}_q$ as shorthand for $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ as shorthand for $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ for brevity. In other words, we want to optimize a $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ that matches $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, which from our derived Equation $\underline{53}$, takes the form:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{58}$$

As $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ also conditions on $\boldsymbol{x}_t$, we can match $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ closely by setting it to the following form:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\bar{\alpha}_t} \tag{59}$$

Variational Diffusion Models

$$= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right]$$

where we have written $\boldsymbol{\mu}_q$ as shorthand for $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ as shorthand for $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ for brevity. In other words, we want to optimize a $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ that matches $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, which from our derived Equation $\underline{53}$, takes the form:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{58}$$

As $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ also conditions on $\boldsymbol{x}_t$, we can match $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ closely by setting it to the following form:

where $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is parameterized by a neural network

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\bar{\alpha}_t} \tag{59}$$

Variational Diffusion Models

$$= \arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right] = \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right\|_2^2\right]$$

where we have written $\boldsymbol{\mu}_q$ as shorthand for $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ as shorthand for $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ for brevity. In other words, we want to optimize a $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ that matches $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, which from our derived Equation $\underline{53}$, takes the form:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{58}$$

As $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ also conditions on $\boldsymbol{x}_t$, we can match $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ closely by setting it to the following form:

where $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is parameterized by a neural network

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\bar{\alpha}_t} \tag{59}$$

Variational Diffusion Models

$$\underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,||\, p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

Therefore, optimizing a VDM boils down to learning a neural network to predict the original ground truth image from an arbitrarily noisified version of it [2]. Furthermore, minimizing the summation term of our derived ELBO objective (Equation 38) across all noise levels can be approximated by minimizing the expectation over all timesteps:

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\mathcal{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t))\right] \\
&= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U\{2,T\}} \left[\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right\|_2^2\right]\right]\right]
\end{aligned}$$

which can then be optimized using stochastic samples over timesteps.

Variational Diffusion Models

Three Equivalent Interpretations

Firstly, we can utilize the reparameterization trick. In our derivation of the form of $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$, we can rearrange Equation $\underline{49}$ to show that:

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \tag{71}$$

Plugging this into our previously derived true denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, we can derive the following alternate parameterization [2]:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{72}$$

Variational Diffusion Models

Three Equivalent Interpretations

Firstly, we can utilize the reparameterization trick. In our derivation of the form of $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$, we can rearrange Equation $\underline{49}$ to show that:

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \tag{71}$$

Plugging this into our previously derived true denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, we can derive the following alternate parameterization [2]:

$$\begin{aligned}
\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} && (72)\\
&= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 && (73)
\end{aligned}$$

Variational Diffusion Models

Three Equivalent Interpretations

Firstly, we can utilize the reparameterization trick. In our derivation of the form of $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$, we can rearrange Equation $\underline{49}$ to show that:

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \tag{71}$$

Plugging this into our previously derived true denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, we can derive the following alternate parameterization [2]:

$$\begin{aligned}
\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} && (72)\\
&= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 && (73)
\end{aligned}$$

Therefore, we can set our approximate denoising transition mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

Variational Diffusion Models

Three Equivalent Interpretations

Firstly, we can utilize the reparameterization trick. In our derivation of the form of $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$, we can rearrange Equation $\underline{49}$ to show that:

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \tag{71}$$

Plugging this into our previously derived true denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, we can derive the following alternate parameterization [2]:

$$\begin{aligned}\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} &(72)\\ &= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 &(73)\end{aligned}$$

Therefore, we can set our approximate denoising transition mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

Here, $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is a neural network that learns to predict the source noise $\boldsymbol{\epsilon}_0$

Variational Diffusion Models

Three Equivalent Interpretations

Firstly, we can utilize the reparameterization trick. In our derivation of the form of $q(\boldsymbol{x}$ can rearrange Equation $\underline{49}$ to show that:

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}$$

Plugging this into our previously derived true denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, derive the following alternate parameterization [2]:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{72}$$

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0 \tag{73}$$

Therefore, we can set our approximate denoising transition mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

$$\arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q(t)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t)\right)\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right]$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right\|_2^2\right]$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left[\left\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right\|_2^2\right]$$

Here, $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is a neural network that learns to predict the source noise $\boldsymbol{\epsilon}_0$

Variational Diffusion Models

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: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
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** $\mathbf{x}_0$

Variational Diffusion Models

Mathematically, for a Gaussian variable $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, Tweedie's Formula states that:

$$\mathbb{E}\left[\boldsymbol{\mu}_z \mid \boldsymbol{z}\right] = \boldsymbol{z} + \boldsymbol{\Sigma}_z \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$$

In this case, we apply it to predict the true posterior mean of $\boldsymbol{x}_t$ given its samples. From Equation $\underline{50}$, we know that:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} \mid \boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

Slides made from:

Tweedie's Formula and Selection Bias

Bradley Efron*
Stanford University

Tweedie's formula

2 Tweedie's formula

Robbins (1956) presents Tweedie's formula as an exponential family generalization of (1.2),

$$\eta \sim g(\cdot) \quad \text{and} \quad z|\eta \sim f_\eta(z) = e^{\eta z - \psi(\eta)} f_0(z). \tag{2.1}$$

Here η is the natural or canonical parameter of the family, $\psi(\eta)$ the cumulant generating function or cgf (which makes $f_\eta(z)$ integrate to 1), and $f_0(z)$ the density when $\eta = 0$. The choice $f_0(z) = \varphi_\sigma(z)$ (1.3), i.e., f_0 a $\mathcal{N}(0, \sigma^2)$ density, yields the normal translation family $\mathcal{N}(\mu, \sigma^2)$, with $\eta = \mu/\sigma^2$. In this case $\psi(\eta) = \frac{1}{2}\sigma^2\eta^2$.

Tweedie's formula

2 Tweedie's formula

Robbins (1956) presents Tweedie's formula as an exponential family generalization of (1.2),

$$\eta \sim g(\cdot) \quad \text{and} \quad z|\eta \sim f_\eta(z) = e^{\eta z - \psi(\eta)} f_0(z). \tag{2.1}$$

Here η is the natural or canonical parameter of the family, $\psi(\eta)$ the cumulant generating function or cgf (which makes $f_\eta(z)$ integrate to 1), and $f_0(z)$ the density when $\eta = 0$. The choice $f_0(z) = \varphi_\sigma(z)$ (1.3), i.e., f_0 a $\mathcal{N}(0, \sigma^2)$ density, yields the normal translation family $\mathcal{N}(\mu, \sigma^2)$, with $\eta = \mu/\sigma^2$. In this case $\psi(\eta) = \frac{1}{2}\sigma^2\eta^2$.

Bayes rule provides the posterior density of η given z,

$$g(\eta|z) = f_\eta(z)g(\eta)/f(z) \tag{2.2}$$

where $f(z)$ is the marginal density

$$f(z) = \int_{\mathcal{Z}} f_\eta(z)g(\eta)\, d\eta, \tag{2.3}$$

Tweedie's formula

$$\eta \sim g(\cdot) \quad \text{and} \quad z|\eta \sim f_\eta(z) = e^{\eta z - \psi(\eta)} f_0(z). \tag{2.1}$$

$$g(\eta|z) = f_\eta(z) g(\eta) / f(z) \tag{2.2}$$

$$f(z) = \int_{\mathcal{Z}} f_\eta(z) g(\eta)\, d\eta, \tag{2.3}$$

Tweedie's formula

$$\eta \sim g(\cdot) \quad \text{and} \quad z|\eta \sim f_\eta(z) = e^{\eta z - \psi(\eta)} f_0(z). \tag{2.1}$$

$$g(\eta|z) = f_\eta(z) g(\eta) / f(z) \tag{2.2}$$

$$f(z) = \int_{\mathcal{Z}} f_\eta(z) g(\eta)\, d\eta, \tag{2.3}$$

$$g(\eta|z) = e^{z\eta - \lambda(z)} \left[g(\eta) e^{-\psi(\eta)} \right] \qquad \text{where } \lambda(z) = \log\left(\frac{f(z)}{f_0(z)} \right); \tag{2.4}$$

(2.4) represents an exponential family with canonical parameter z and cgf $\lambda(z)$.

Tweedie's formula

$$\eta \sim g(\cdot) \quad \text{and} \quad z|\eta \sim f_\eta(z) = e^{\eta z - \psi(\eta)} f_0(z). \tag{2.1}$$

$$g(\eta|z) = f_\eta(z) g(\eta) / f(z) \tag{2.2}$$

$$f(z) = \int_{\mathcal{Z}} f_\eta(z) g(\eta)\, d\eta, \tag{2.3}$$

$$g(\eta|z) = e^{z\eta - \lambda(z)} \left[g(\eta) e^{-\psi(\eta)} \right] \qquad \text{where } \lambda(z) = \log\left(\frac{f(z)}{f_0(z)} \right); \tag{2.4}$$

(2.4) represents an exponential family with canonical parameter z and cgf $\lambda(z)$. Differentiating $\lambda(z)$ yields the posterior cumulants of η given z,

$$E\{\eta|z\} = \lambda'(z), \qquad \text{var}\{\eta|z\} = \lambda''(z), \tag{2.5}$$

Tweedie's formula

Letting

$$l(z) = \log(f(z)) \quad \text{and} \quad l_0(z) = \log(f_0(z)) \tag{2.6}$$

we can express the posterior mean and variance of $\eta|z$ as

$$\eta|z \sim \left(l'(z) - l_0'(z), l''(z) - l_0''(z)\right). \tag{2.7}$$

Tweedie's formula

Letting

$$l(z) = \log(f(z)) \quad \text{and} \quad l_0(z) = \log(f_0(z)) \tag{2.6}$$

we can express the posterior mean and variance of $\eta|z$ as

$$\eta|z \sim \left(l'(z) - l_0'(z), l''(z) - l_0''(z)\right). \tag{2.7}$$

In the normal translation family $z \sim \mathcal{N}(\mu, \sigma^2)$ (having $\mu = \sigma^2\eta$), (2.7) becomes

$$\mu|z \sim \left(z + \sigma^2 l'(z), \sigma^2\left(1 + \sigma^2 l''(z)\right)\right). \tag{2.8}$$

Tweedie's formula

Letting

$$l(z) = \log(f(z)) \quad \text{and} \quad l_0(z) = \log(f_0(z)) \tag{2.6}$$

we can express the posterior mean and variance of $\eta|z$ as

$$\eta|z \sim \left(l'(z) - l_0'(z), l''(z) - l_0''(z)\right). \tag{2.7}$$

In the normal translation family $z \sim \mathcal{N}(\mu, \sigma^2)$ (having $\mu = \sigma^2\eta$), (2.7) becomes

$$\mu|z \sim \left(z + \sigma^2 l'(z), \sigma^2\left(1 + \sigma^2 l''(z)\right)\right). \tag{2.8}$$

The unbiased estimate of μ for $z \sim \mathcal{N}(\mu, \sigma^2)$ is z itself, so we can write (1.4), or (2.8), in a form emphasized in Section 4,

$$E\{\mu|z\} = \text{ unbiased estimate plus Bayes correction.} \tag{2.9}$$

Tweedie's formula

Letting

$$l(z) = \log(f(z)) \quad \text{and} \quad l_0(z) = \log(f_0(z)) \tag{2.6}$$

we can express the posterior mean and variance of $\eta|z$ as

$$\eta|z \sim \left(l'(z) - l_0'(z), l''(z) - l_0''(z)\right). \tag{2.7}$$

In the normal translation family $z \sim \mathcal{N}(\mu, \sigma^2)$ (having $\mu = \sigma^2\eta$), (2.7) becomes

$$\mu|z \sim \left(z + \sigma^2 l'(z), \sigma^2\left(1 + \sigma^2 l''(z)\right)\right). \tag{2.8}$$

The unbiased estimate of μ for $z \sim \mathcal{N}(\mu, \sigma^2)$ is z itself, so we can write (1.4), or (2.8), in a form emphasized in Section 4,

$$E\{\mu|z\} = \text{ unbiased estimate plus Bayes correction.} \tag{2.9}$$

$$\mathbb{E}\left[\boldsymbol{\mu}_z \mid \boldsymbol{z}\right] = \boldsymbol{z} + \boldsymbol{\Sigma}_z \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$$

Back to:

Understanding Diffusion Models: A Unified Perspective

An intuitive, accessible tutorial on diffusion models.

AUTHORS

Calvin Luo

AFFILIATIONS

Google Research [3]

Brown University

PUBLISHED

Aug. 26, 2022

https://www.calvinyluo.com/2022/08/26/diffusion-tutorial.html

Variational Diffusion Models

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} \mid \boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

Variational Diffusion Models

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} \mid \boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

where we write $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ as $\nabla \log p(\boldsymbol{x}_t)$ for notational simplicity. According to Tweedie's Formula, the best estimate for the true mean that $\boldsymbol{x}_t$ is generated from, $\boldsymbol{\mu}_{x_t} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$, is defined as:

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)$$

$$\therefore \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

Variational Diffusion Models

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$$

Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} \mid \boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

where we write $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ as $\nabla \log p(\boldsymbol{x}_t)$ for notational simplicity. According to Tweedie's Formula, the best estimate for the true mean that $\boldsymbol{x}_t$ is generated from, $\boldsymbol{\mu}_{x_t} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$, is defined as:

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)$$

$$\therefore \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

Then, we can plug Equation 79 into our previousy derived ground-truth denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ once again and derive a new parameterization:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\boldsymbol{x}_0}{1 - \bar{\alpha}_t} \tag{80}$$

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1 - \alpha_t}{\sqrt{\alpha_t}}\nabla \log p(\boldsymbol{x}_t) \tag{81}$$

Variational Diffusion Mode

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} \mid \boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q(t)\right) \| \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t)\right)\right) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left[\left\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{\alpha_t}\left[\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p(\boldsymbol{x}_t)\right\|_2^2\right]
\end{aligned}$$

where we write $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ as $\nabla \log p(\boldsymbol{x}_t)$ for notational simplicity. According to Tweedie's Formula, the best estimate for the true mean that $\boldsymbol{x}_t$ is generated from, $\boldsymbol{\mu}_{x_t} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$, is defined as:

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)$$

$$\therefore \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

Then, we can plug Equation 79 into our previousy derived ground-truth denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ once again and derive a new parameterization:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{80}$$

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla \log p(\boldsymbol{x}_t) \tag{81}$$

Variational Diffusion Mode

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t} \mid \boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$$

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q(t)\right) \,||\, \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t)\right)\right) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left[\left\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left[\left\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right\|_2^2\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{\alpha_t}\left[\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p(\boldsymbol{x}_t)\right\|_2^2\right]
\end{aligned}$$

where we write $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ as $\nabla \log p(\boldsymbol{x}_t$ Formula, the best estimate for the true mean that $\boldsymbol{x}_t$ is generated from, $\boldsymbol{\mu}_{x_t} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$, is defined as:

Here, $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is a neural network that learns to predict the score function $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)$$

$$\therefore \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

Then, we can plug Equation 79 into our previousy derived ground-truth denoising transition mean $\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ once again and derive a new parameterization:

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \tag{80}$$

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla \log p(\boldsymbol{x}_t) \tag{81}$$

Score-based Generative Models

To begin to understand why optimizing a score function makes sense, we take a detour and revisit energy-based models [14, 15]. Arbitrarily flexible probability distributions can be written

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{Z_{\boldsymbol{\theta}}} e^{-f_{\boldsymbol{\theta}}(\boldsymbol{x})} \tag{88}$$

One way to avoid calculating or modeling the normalization constant is by using a neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})$ to learn the score function $\nabla \log p(\boldsymbol{x})$ of distribution $p(\boldsymbol{x})$ instead. This is motivated by the observation that taking the derivative of the log of both sides of Equation 88

$$\begin{aligned}
\nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} \log\left(\frac{1}{Z_{\boldsymbol{\theta}}} e^{-f_{\boldsymbol{\theta}}(\boldsymbol{x})}\right) && (89)\\
&= \nabla_{\boldsymbol{x}} \log \frac{1}{Z_{\boldsymbol{\theta}}} + \nabla_{\boldsymbol{x}} \log e^{-f_{\boldsymbol{\theta}}(\boldsymbol{x})} && (90)\\
&= -\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x}) && (91)\\
&\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) && (92)
\end{aligned}$$

network can then be optimized by minimizing the Fisher Divergence between the estimated score and the ground truth score function:

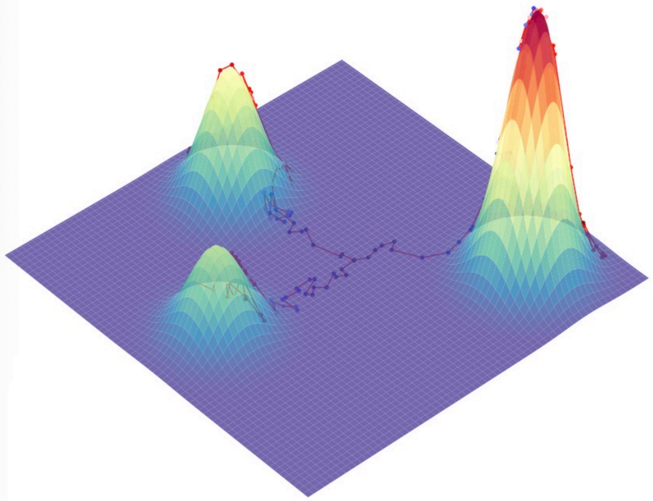
$$\mathbb{E}_{p(\boldsymbol{x})}\left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla \log p(\boldsymbol{x})\|_2^2\right] \tag{93}$$

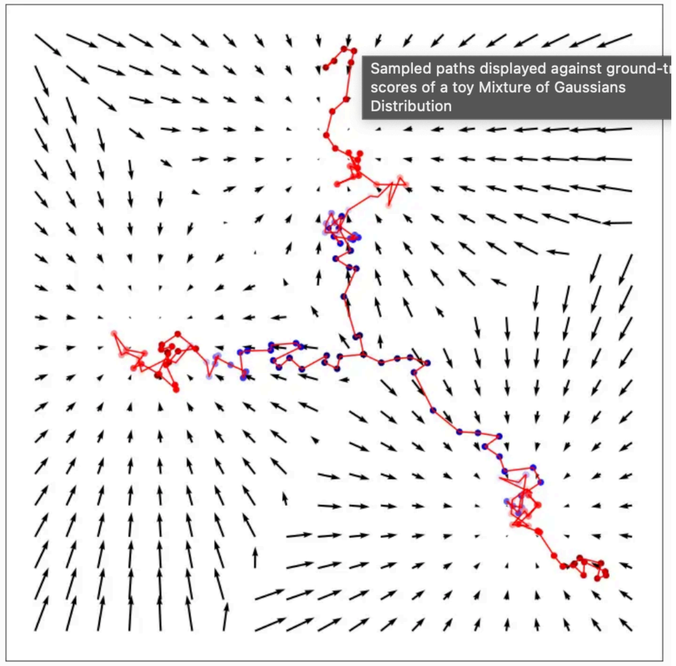
Score-based Generative Models

network can then be optimized by minimizing the Fisher Divergence between the estimated score and the ground truth score function:

$$\mathbb{E}_{p(\boldsymbol{x})}\left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla \log p(\boldsymbol{x})\|_2^2\right] \tag{93}$$

What does the score function represent? For every $\boldsymbol{x}$, taking the gradient of its log likelihood with respect to $\boldsymbol{x}$ essentially describes what direction in data space to move in order to further increase its likelihood. Intuitively, then, the score function defines a vector field over the entire space that data $\boldsymbol{x}$ inhabits, pointing towards the modes. Visually, this is depicted by the black arrows in the right plot of the figure below.



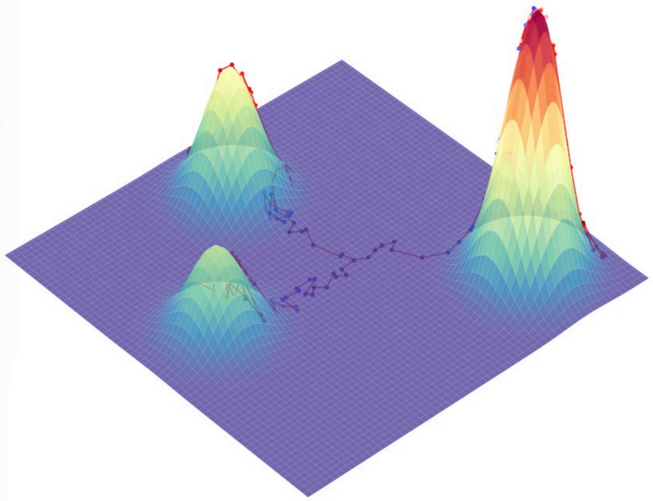


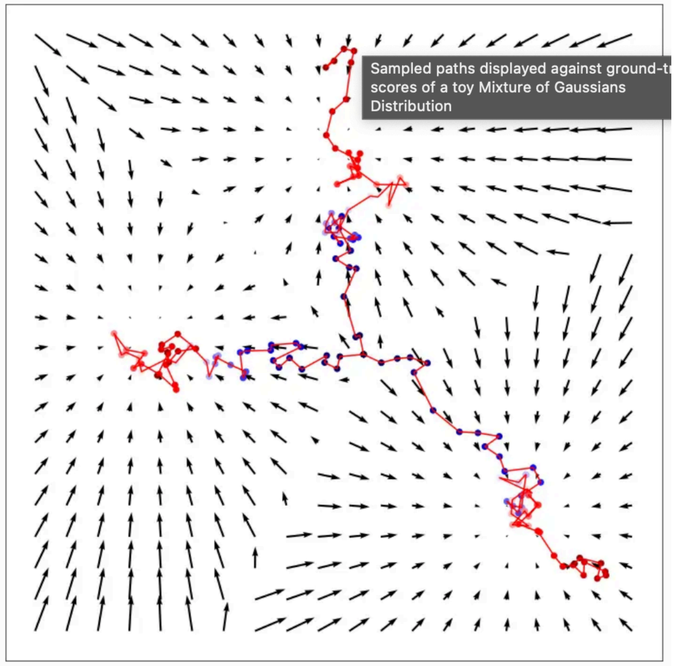
Score-based Generative Models

network can then be optimized by minimizing the Fisher Divergence between the estimated score and the ground truth score function:

$$\mathbb{E}_{p(\boldsymbol{x})}\left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla \log p(\boldsymbol{x})\|_2^2\right] \tag{93}$$

What does the score function represent? For every $\boldsymbol{x}$, taking the gradient of its log likelihood with respect to $\boldsymbol{x}$ essentially describes what direction in data space to move in order to further increase its likelihood. Intuitively, then, the score function defines a vector field over the entire space that data $\boldsymbol{x}$ inhabits, pointing towards the modes. Visually, this is depicted by the black arrows in the right plot of the figure below.





Then, by learning the score function of the true data distribution, we can generate samples by starting at any arbitrary point in the same space and iteratively following the score until a mode is reached. This sampling procedure is known as Langevin dynamics, and is mathematically described as:

$$\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + c\nabla \log p(\boldsymbol{x}_i) + \sqrt{2c}\boldsymbol{\epsilon}, \quad i = 0, 1, \ldots, K \tag{94}$$

where $\boldsymbol{x}_0$ is randomly sampled from a prior distribution (such as uniform), and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$ is

Connection to SDEs

Therefore, we have established an explicit connection between Variational Diffusion Models and Score-based Generative Models, both in their training objectives and sampling procedures.

One question is how to naturally generalize diffusion models to an infinite number of timesteps. Under the Markovian HVAE view, this can be interpreted as extending the number of hierarchies to infinity $T \to \infty$. It is clearer to represent this from the equivalent score-based generative model perspective; under an infinite number of noise scales, the perturbation of an image over continuous time can be represented as a stochastic process, and therefore described by a stochastic differential equation (SDE). Sampling is then performed by reversing the SDE, which naturally requires estimating the score function at each continuous-valued noise level [12].

Connection to SDEs

Variational Diffusion Models

Diederik P. Kingma*, **Tim Salimans***, **Ben Poole, Jonathan Ho**
Google Research

5 Continuous-time model: $T \to \infty$

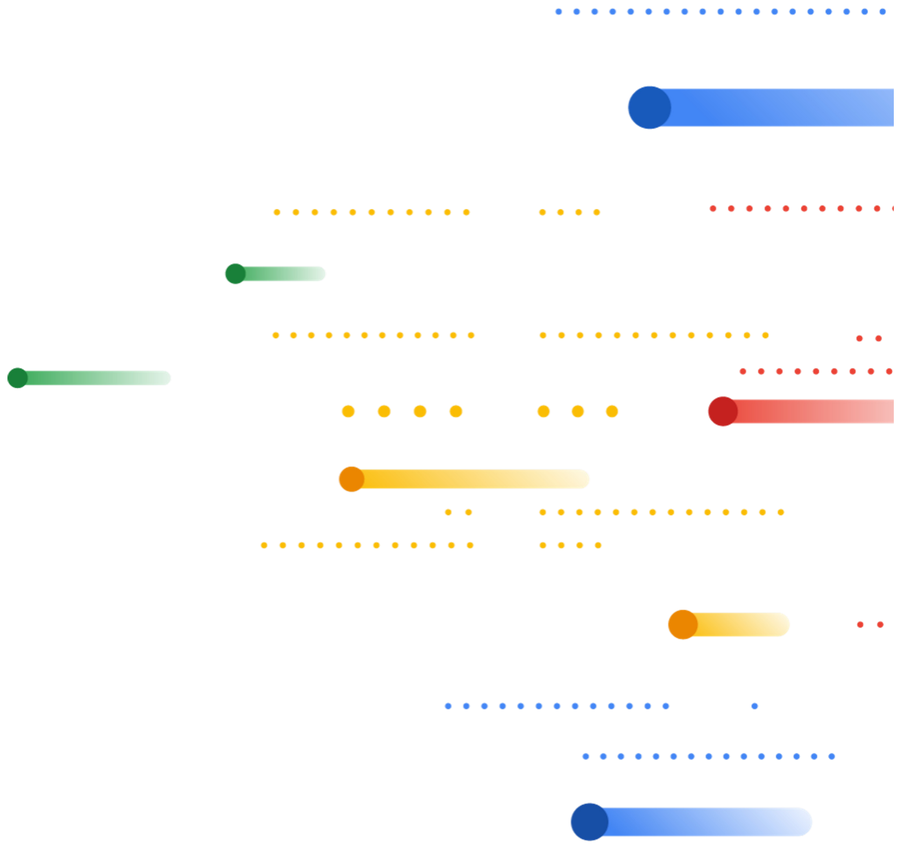
Since taking more time steps leads to a better VLB, we now take $T \to \infty$, effectively treating time t as continuous rather than discrete. The model for $p(\mathbf{z}_t)$ can in this case be described as a continuous time diffusion process [Song et al., 2021b] governed by a stochastic differential equation;

$$\mathcal{L}_\infty(\mathbf{x}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),t\sim\mathcal{U}(0,1)}\left[\gamma'_{\boldsymbol{\eta}}(t)\,\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)\|_2^2\right]$$

Back to: Google DeepMind

Variational Autoencoders and Diffusion Models

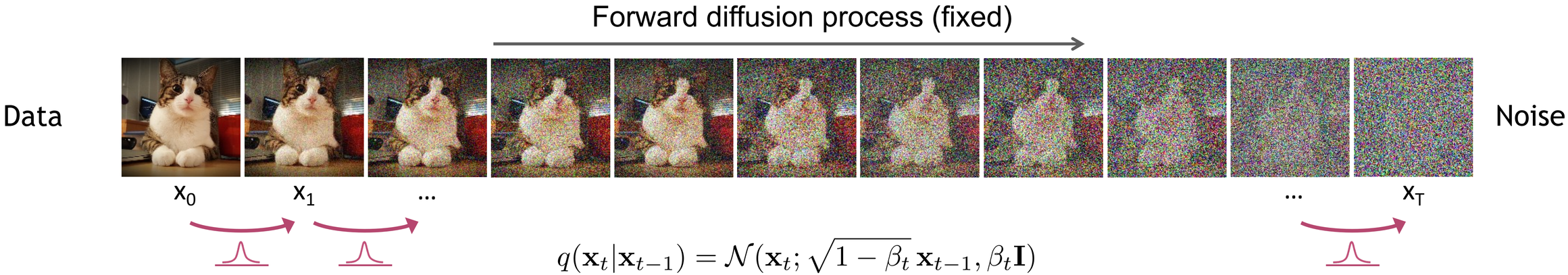
Ruiqi Gao @Stanford cs231n
May 25, 2023



Continuous-time diffusion models

Stochastic differential equation framework

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



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

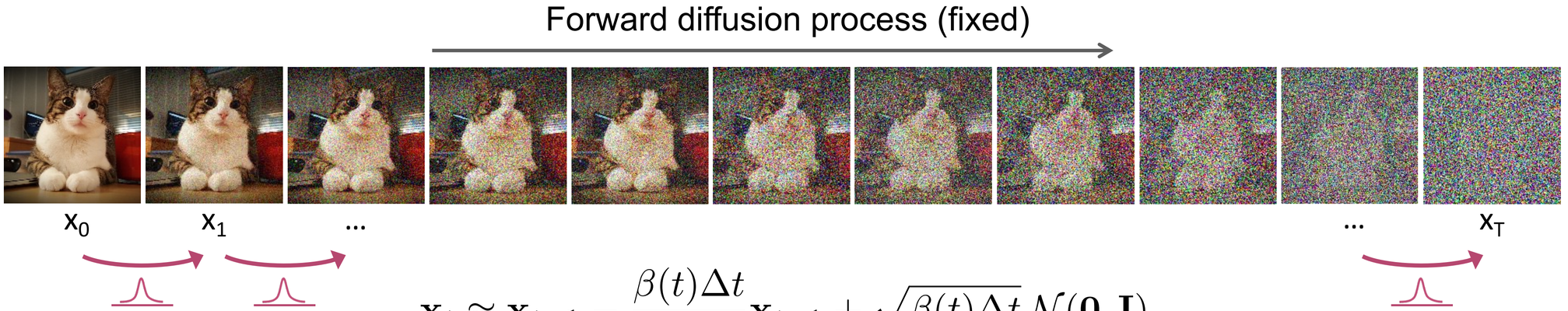
$$\begin{aligned}
\mathbf{x}_t &= \sqrt{1-\beta_t}\,\mathbf{x}_{t-1} + \sqrt{\beta_t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) \\
&= \sqrt{1-\beta(t)\Delta t}\,\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0},\mathbf{I}) && (\beta_t := \beta(t)\Delta t) \\
&\approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta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

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Forward Diffusion Process as Stochastic Differential Equation

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



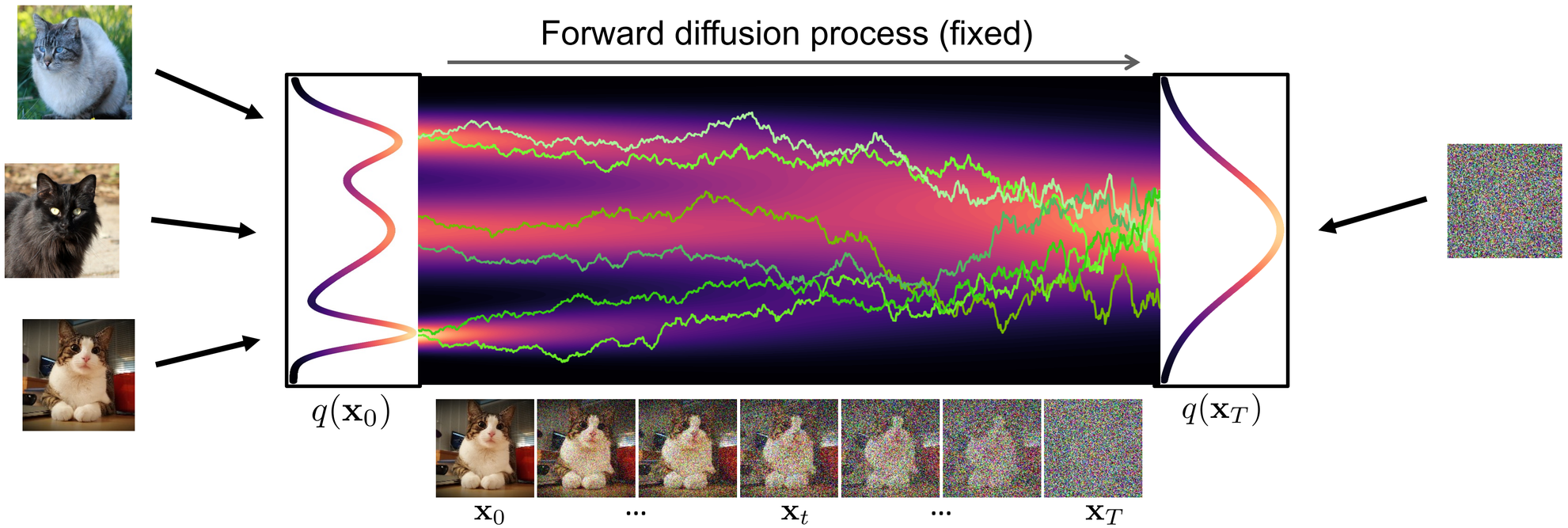
$$\mathbf{x}_t \approx \mathbf{x}_{t-1} - \frac{\beta(t)\Delta t}{2}\mathbf{x}_{t-1} + \sqrt{\beta(t)\Delta t}\,\mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Stochastic Differential Equation (SDE)
describing the diffusion in infinitesimal limit

Song et al., "Score-Based Generative Modeling through Stochastic Differential Equations", *ICLR*, 2021
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Forward Diffusion Process as Stochastic Differential Equation



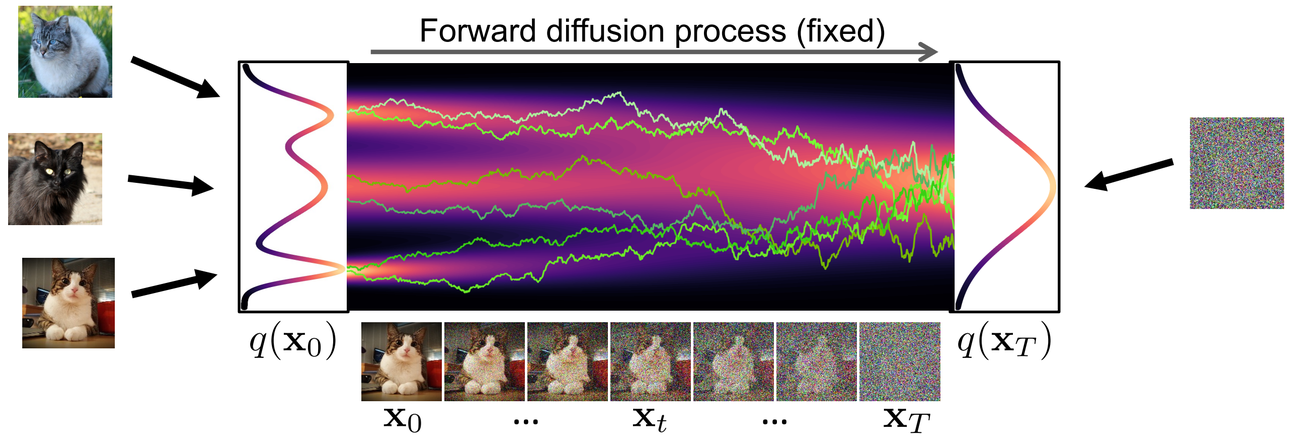
Forward Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = \underbrace{-\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t}_{\text{drift term (pulls towards mode)}} + \underbrace{\sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t}_{\text{diffusion term (injects noise)}}$$

Song et al., *ICLR*, 2021

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

The Generative Reverse Stochastic Differential Equation



Forward Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

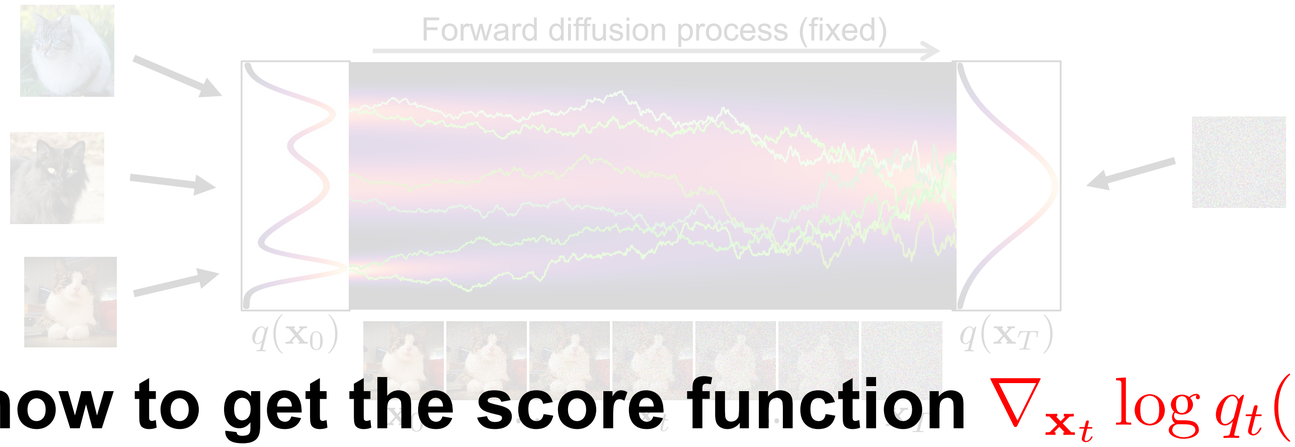
Reverse Generative Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

➡ **Simulate reverse diffusion process: Data generation from random noise!**

Song et al., *ICLR*, 2021
Anderson, in *Stochastic Processes and their Applications*, 1982
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

The Generative Reverse Stochastic Differential Equation



But how to get the score function $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$?

Forward Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

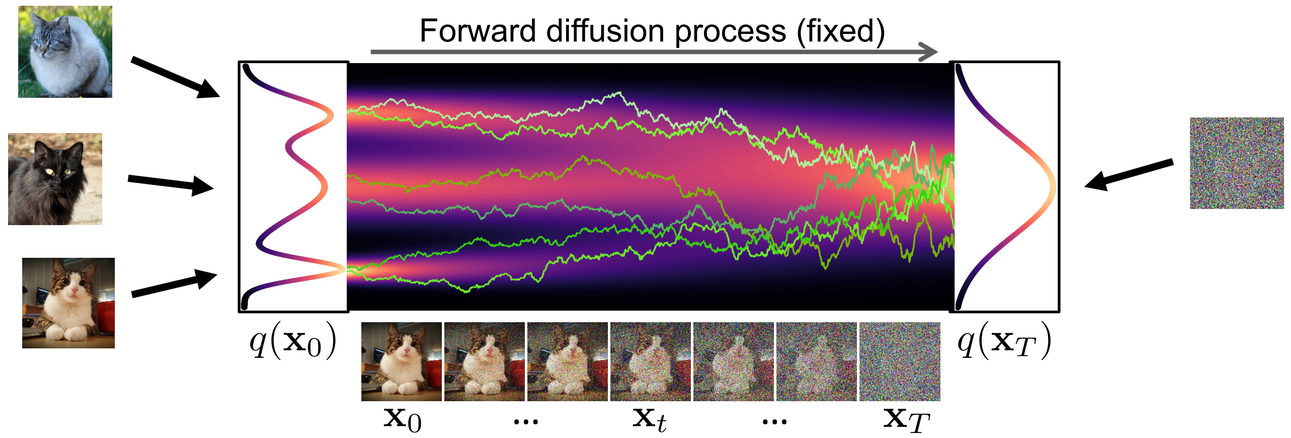
Reverse Generative Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = \underbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}_{\text{drift term}}\mathrm{d}t + \underbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}_{\text{diffusion term}}$$

➡ **Simulate reverse diffusion process: Data generation from random noise!**

Song et al., *ICLR*, 2021
Anderson, in *Stochastic Processes and their Applications*, 1982

Score Matching



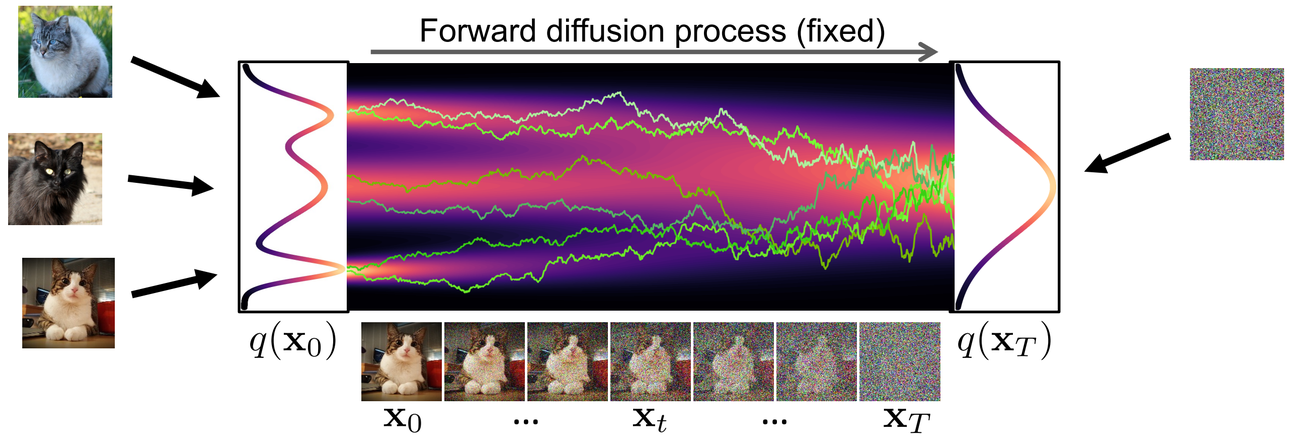
- Naïve idea, learn model for the score function by direct regression?

$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\text{diffusion time } t} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)}}_{\text{diffused data } \mathbf{x}_t} \underbrace{\tilde{w}(t)}_{\text{weighting function}} \cdot \underbrace{||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\text{neural network}} - \underbrace{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)||_2^2}_{\text{score of diffused data (marginal)}}$$

➡ **But $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ (score of the *marginal diffused density* $q_t(\mathbf{x}_t)$) is not tractable!**

Vincent, "A Connection Between Score Matching and Denoising Autoencoders", *Neural Computation*, 2011
Song and Ermon, "Generative Modeling by Estimating Gradients of the Data Distribution", *NeurIPS*, 2019
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Denoising Score Matching



- Instead, diffuse individual data points $\mathbf{x}_0$. Diffused $q_t(\mathbf{x}_t|\mathbf{x}_0)$ ***is*** tractable!
- **Denoising Score Matching:**

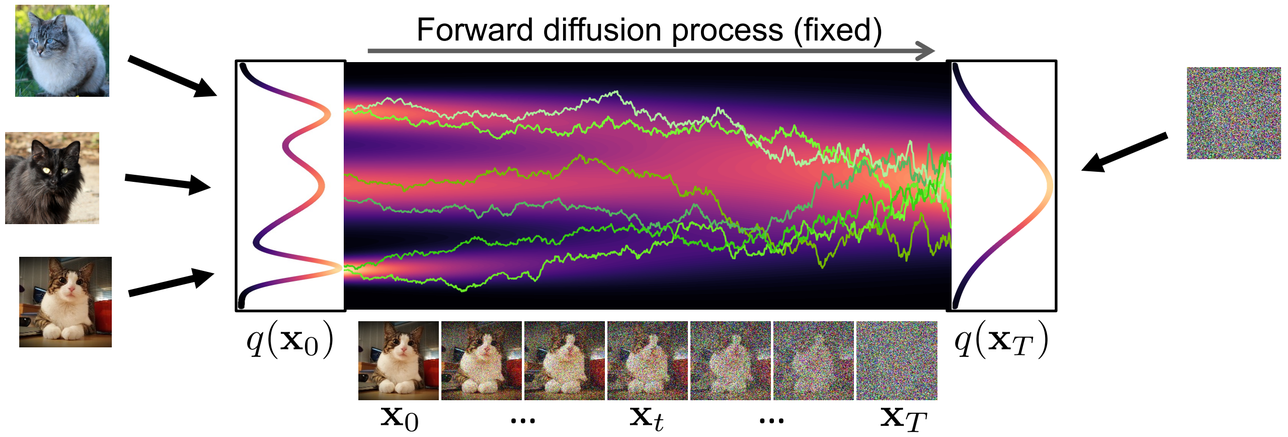
$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\text{diffusion time } t} \underbrace{\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}}_{\text{data sample } \mathbf{x}_0} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)}}_{\text{diffused data sample } \mathbf{x}_t} \underbrace{\tilde{w}(t)}_{\text{weighting function}} \cdot ||\underbrace{\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\text{neural network}} - \underbrace{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)}_{\text{score of diffused data sample}}||_2^2$$

➡ **After expectations,** $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$**!**

Vincent, in *Neural Computation*, 2011
Song and Ermon, *NeurIPS*, 2019
Song et al. *ICLR*, 2021
slide from https://cvpr2022-tutorial-diffusion-models.github.io/

Denoising Score Matching

Epsilon-prediction parametrization



- **Denoising Score Matching:**

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)}\tilde{w}(t)\cdot||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)-\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\mathbf{x}_0)||_2^2$$

  - Re-parametrized sampling: $\mathbf{x}_t=\alpha_t\mathbf{x}_0+\sigma_t\boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
    - Score function: $\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\mathbf{x}_0)=-\nabla_{\mathbf{x}_t}\frac{(\mathbf{x}_t-\alpha_t\mathbf{x}_0)^2}{2\sigma_t^2}=-\frac{\mathbf{x}_t-\alpha_t\mathbf{x}_0}{\sigma_t^2}=-\frac{\alpha_t\mathbf{x}_0+\sigma_t\boldsymbol{\epsilon}-\alpha_t\mathbf{x}_0}{\sigma_t^2}=-\frac{\boldsymbol{\epsilon}}{\sigma_t}$
    - Neural network model: $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t):=-\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)}{\sigma_t}$

Vincent, in *Neural Computation*, 2011
Song and Ermon, *NeurIPS*, 2019
Song et al. *ICLR*, 2021

➡ $$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\hat{w}(t)\cdot||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)||_2^2 \qquad \hat{w}(t)=\frac{\tilde{w}(t)}{\sigma_t}$$

slide from https://cvpr2022-tutorial-diffusion-models.github.io/

What is the ELBO for continuous-time diffusion models?

- Denoising Score Matching:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\hat{w}(t)\cdot||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)||_2^2$$

- [Kingma et al, 2021] showed that the variational upper bound (negative ELBO) can be reduced to a simple variant:

$$-\text{ELBO}(\theta) = \frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}-\frac{d\lambda}{dt}\cdot||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)||_2^2+\text{const}$$

$$\text{where } \mathbf{x}_t = \alpha_t\mathbf{x}_0+\sigma_t\boldsymbol{\epsilon}$$

$$\lambda = \log\frac{\alpha_t^2}{\sigma_t^2} \quad \text{(signal-to-noise ratio of timestep } t\text{)}$$

- In practice, different loss weightings trade off between model with good perceptual quality vs. high log-likelihood:
  - Perceptual quality, e.g., $\hat{w}(t)=1$
  - High likelihood: -ELBO

➡ How to explain such discrepancy?
Can we still trust ELBO / maximum likelihood?

Kingma et al., “Variational diffusion models”, NeurIPS 2021.

Weighted diffusion objectives

Understanding diffusion objectives as the ELBO with data augmentation

- Summarize different types of diffusion objectives as the following **weighted diffusion objective**:

$$\mathcal{L}_w(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[w(t)\cdot -\frac{d\lambda}{dt}\cdot ||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)||_2^2\right]$$

- [Kingma & Gao, 2023] showed that if $w(t)$ is a monotonically increasing function, then the weighted diffusion objective is equivalent to the **negative ELBO with data augmentation** (Gaussian addition noise):

$$\mathcal{L}_w(\boldsymbol{\theta}) \geq \underbrace{\mathbb{E}_{t\sim p_w(t)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)}\left[-\log p(\mathbf{x}_t)\right]}_{\text{Neg. log lik. of noise-perturbed data}} + \text{const}$$

$$\text{where } p_w(t) \propto \frac{dw(t)}{dt}$$

- Therefore, the ELBO objective is compatible with perceptual quality when combined with simple data augmentation (additive noise)

Kingma & Gao, “Understanding Diffusion Objectives\\as the ELBO with Data Augmentation”, 2023.