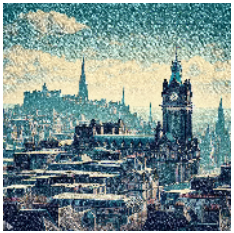
Advanced Topics in Machine Learning (deep generative modelling)

Lecture 8: Diffusion models I

Nikolay Malkin

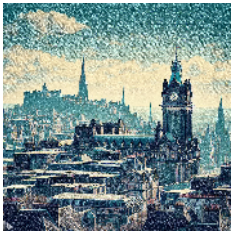
10 March 2026

Dynamics-based generative models in practice



'Edinburgh from Calton Hill, pointillist style'

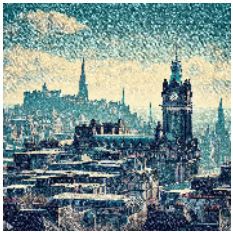
Dynamics-based generative models in practice



'Edinburgh from Calton Hill, pointillist style'

[Janner et al., ICML'22]

Dynamics-based generative models in practice

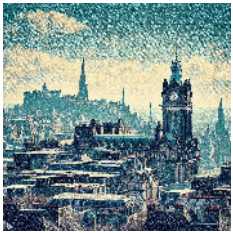


'Edinburgh from Calton Hill, pointillist style'

[Graikos et al., NeurIPS'22]

[Janner et al., ICML'22]

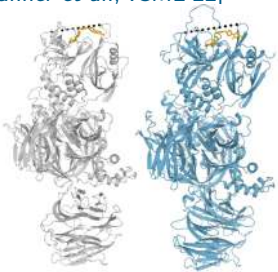
Dynamics-based generative models in practice



'Edinburgh from Calton Hill, pointillist style'

[Graikos et al., NeurIPS'22]

[Janner et al., ICML'22]



AlphaFold 3

Dynamics-based generative models in practice

[Zhang and Gienger, 2024]

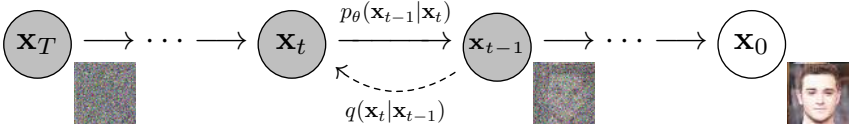
Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

- **Lecture 8:** Diffusion models are hierarchical latent variable models / deep VAEs



[Ho et al., 'Denoising diffusion probabilistic models']

Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

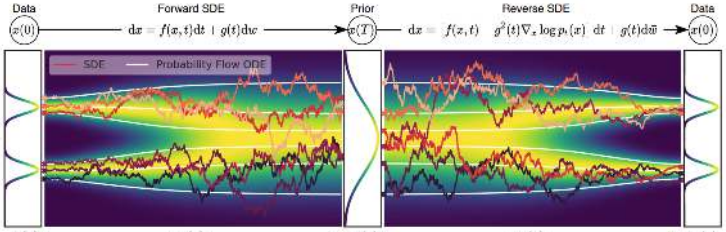
- **Lecture 8:** Diffusion models are hierarchical latent variable models / deep VAEs
- **Lecture 9:** Diffusion models are score-based models

[Song et al., 'Generative modeling by estimating...' blog post]

Outline of weeks 8-10

What is a diffusion model, how to understand it from different perspectives, and where to use it in practice:

- **Lecture 8:** Diffusion models are hierarchical latent variable models / deep VAEs
- **Lecture 9:** Diffusion models are score-based models
- **Lecture 10:** Diffusion models are continuous-time processes



[Song et al., 'Score-based generative modeling. . . ']

Outline of Lecture 8

- Historical overview
- Review of latent variable models
- Diffusion models as hierarchical generative models
  - Evaluation and likelihood estimation
  - Latent variable model perspective

- Historical overview
- Review of latent variable models
- Diffusion models as hierarchical generative models
  - Evaluation and likelihood estimation
  - Latent variable model perspective

Historical overview

The principles have not changed much.

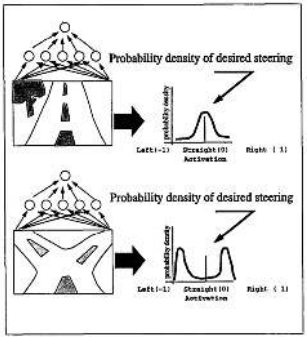
Historical overview

- Early ‘90s: proposals to model dynamics in neural activations
  - Combined two ideas from statistical physics: neural nets / connectionist models and diffusion processes / SDEs

COGNITIVE SCIENCE **17,** 463–496 (1993)

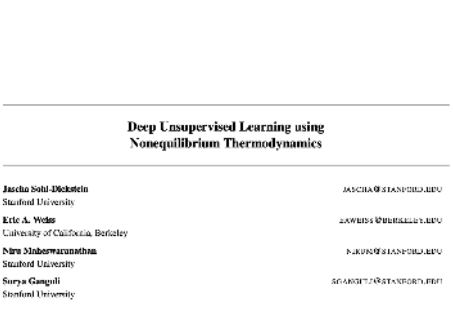
Learning Continuous Probability Distributions with Symmetric Diffusion Networks

JAVIER R. MOVELLAN
University of California, San Diego
JAMES L. MCCLELLAND
Carnegie Mellon University



Historical overview

- deep learning revolution (better hardware and algorithms) ...
- 2015: first deep generative model based on diffusion



Deep Unsupervised Learning using
Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein
Stanford University
JASCHA@STANFORD.EDU

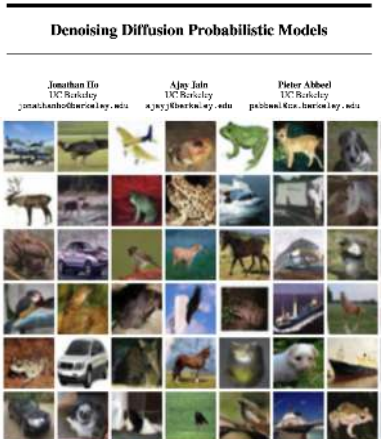
Eric A. Weiss
University of California, Berkeley
EAWEISS@BERKELEY.EDU

Niru Maheswaranathan
Stanford University
NIRUM@STANFORD.EDU

Surya Ganguli
Stanford University
SGANGULI@STANFORD.EDU

Historical overview

- 2020–2021: new training techniques and architectures make diffusion models competitive



Historical overview

- 2020–2021: new training techniques and architectures make diffusion models competitive

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal*
OpenAI
prafulla@openai.com

Alex Nichol*
OpenAI
alex@openai.com

Historical overview

- Early ‘90s: proposals to model dynamics in neural activations
  - Combined two ideas from statistical physics: neural nets / connectionist models and diffusion processes / SDEs
- deep learning revolution (better hardware and algorithms) ...
- 2015: first deep generative model based on diffusion
- 2020–2021: new training techniques and architectures make diffusion models competitive

Historical overview

- Early ‘90s: proposals to model dynamics in neural activations
  - Combined two ideas from statistical physics: neural nets / connectionist models and diffusion processes / SDEs
- deep learning revolution (better hardware and algorithms) . . .
- 2015: first deep generative model based on diffusion
- 2020–2021: new training techniques and architectures make diffusion models competitive
- 2022–: continued improvements in modeling, efficiency, and applications

Historical overview

Discrete, compositional, and symbolic representations through attractor dynamics

Andrew J. Nam[*]
Princeton University
Princeton, NJ
andrewnam@princeton.edu

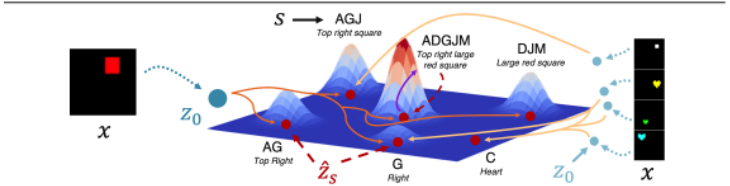
Eric Elmoznino
Mila, Université de Montréal
Montréal, QC, Canada
eric.elmoznino@mila.quebec

Nikolay Malkin[†]
University of Edinburgh
Edinburgh, Scotland, UK
nmalkin@inf.ed.ac.uk

James L. McClelland
Stanford University
Stanford, CA
jlmcc@stanford.edu

Yoshua Bengio[‡]
Mila, Université de Montréal
Montréal, QC, Canada
yoshua.bengio@mila.quebec

Guillaume Lajoie[†]
Mila, Université de Montréal
Montréal, QC, Canada
guillaume.lajoie@mila.quebec



- Historical overview
- Review of latent variable models
- Diffusion models as hierarchical generative models
  - Evaluation and likelihood estimation
  - Latent variable model perspective

Latent variable models

Generative models with **latent variables** z: To approximate π_{data} by a model p, model a joint distribution $p_\theta(x, z)$ over observed x and latent z to satisfy

$$p_\theta(x) = \int_z p_\theta(x, z)\, dz \approx \pi_{\text{data}}(x)$$

Latent variable models

Generative models with **latent variables** z: To approximate π_{data} by a model p, model a joint distribution $p_\theta(x, z)$ over observed x and latent z

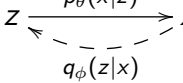
$$z \xrightarrow{p_\theta(x|z)} x \qquad p_\theta(x) = \int_z p_\theta(z) p_\theta(x \mid z)\, dz$$

'Ancestral' sampling of x:

- Sample the latent z from a distribution $p_\theta(z)$.
- Sample x from $p_\theta(x \mid z)$.

Latent variable models

Generative models with **latent variables** z: To approximate π_{data} by a model p, model a joint distribution $p_\theta(x, z)$ over observed x and latent z



$$p_\theta(x) = \int_z p_\theta(z) p_\theta(x \mid z)\, dz$$

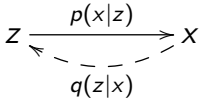
'Ancestral' sampling of x:

- Sample the latent z from a distribution $p_\theta(z)$.
- Sample x from $p_\theta(x \mid z)$.

The posterior over z, by Bayes' rule (we often approximate it by a $q_\phi(z \mid x)$):

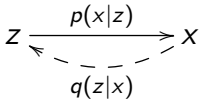
$$p_\theta(z \mid x) = \frac{p_\theta(z) p_\theta(x \mid z)}{p_\theta(x)} \propto p_\theta(z) p_\theta(x \mid z)$$

Latent variables ‘encode’ the data

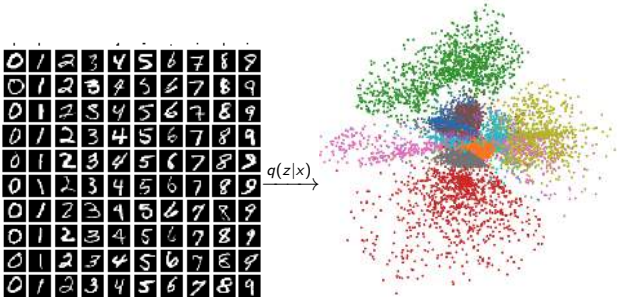


The approximate posterior $q(z \mid x)$ (often a Gaussian in VAEs) stochastically ‘encodes’ the data x into a latent z

Latent variables ‘encode’ the data

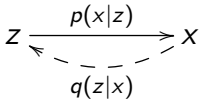


The approximate posterior $q(z \mid x)$ (often a Gaussian in VAEs) stochastically ‘encodes’ the data x into a latent z

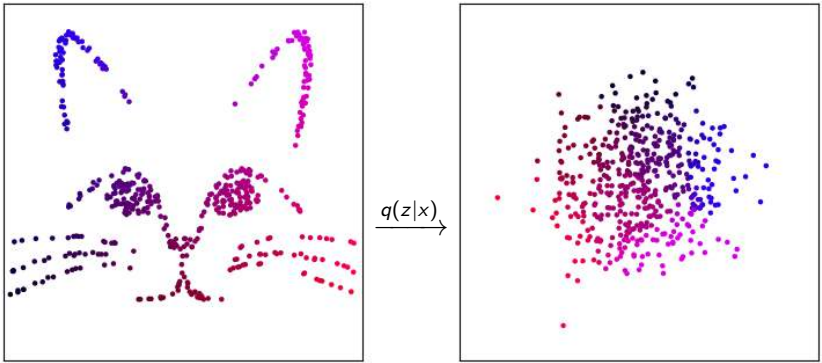


VAE latent space explorer

Latent variables ‘encode’ the data



The approximate posterior $q(z \mid x)$ (often a Gaussian in VAEs) stochastically ‘encodes’ the data x into a latent z



MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i)$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

$$= \log p_\theta(x) - \mathrm{KL}(q_i \,\|\, p_\theta(z|x_i)))$$

with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

$$= \log p_\theta(x) - \mathrm{KL}(q_i \,\|\, p_\theta(z|x_i)))$$

with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)

MLE training of latent variable models

Given a dataset $\{x_i\}_{i=1}^N$, we want to maximise

$$\sum_i \log p_\theta(x_i) = \sum_i \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz$$

w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$.

- **Gibbs' inequality:** for any distribution $q_i(z)$,

$$\log p_\theta(x_i) = \log \int_z p_\theta(z) p_\theta(x_i \mid z)\, dz \geq \underbrace{\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)}\, dz}_{\text{evidence lower bound (ELBO)}}$$

$$= \log p_\theta(x) - \mathrm{KL}(q_i \,\|\, p_\theta(z|x_i)))$$

  with equality when $q_i(z) = p_\theta(z \mid x_i) \propto p_\theta(z) p_\theta(x_i \mid z)$ (true posterior)
- Idea: to maximise $\log p(x^t)$, maximise the ELBO
  - $\rightsquigarrow$ **EM alg.** ("Maximum likelihood from incomplete data" [Dempster et al., 1977])

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z \sim q_i} \left[\log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} \right]$$

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z \sim q_i} \left[\log \frac{p_\theta(z) p_\theta(x_i \mid z)}{q_i(z)} \right]$$

- **(Amortised) variational EM:** approximate q_i with a model $q_\phi(z \mid x_i)$ and train it to approximate the true posterior $p_\theta(z \mid x_i)$ (**E-step**)

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z\sim q_i}\left[\log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)}\right]$$

- **(Amortised) variational EM:** approximate q_i with a model $q_\phi(z \mid x_i)$ and train it to approximate the true posterior $p_\theta(z \mid x_i)$ (**E-step**)
- **(Gradient-based) M-step:** sample $z \sim q_\phi(z \mid x_i)$ and maximise $\log p_\theta(z)p_\theta(x_i \mid z)$ w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$

Gradient-based EM

Idea: to maximise $\log p_\theta(x_i)$, maximise the ELBO

$$\int_z q_i(z) \log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)} = \mathbb{E}_{z\sim q_i}\left[\log \frac{p_\theta(z)p_\theta(x_i \mid z)}{q_i(z)}\right]$$

- **(Amortised) variational EM:** approximate q_i with a model $q_\phi(z \mid x_i)$ and train it to approximate the true posterior $p_\theta(z \mid x_i)$ (**E-step**)
- **(Gradient-based) M-step:** sample $z \sim q_\phi(z \mid x_i)$ and maximise $\log p_\theta(z)p_\theta(x_i \mid z)$ w.r.t. parameters of $p_\theta(z)$ and $p_\theta(x \mid z)$
- Intuitively:
  - E-step: learn to encode x into z that explains it well
  - M-step: for data x, sample explanation z and learn to recover x from z

Hierarchical latent variable models

$$z_N \xrightarrow{p(z_{N-1}|z_N)} z_{N-1} \xrightarrow{p(z_{N-2}|z_{N-1})} \cdots \longrightarrow z_1 \xrightarrow{p(x|z_1)} x$$

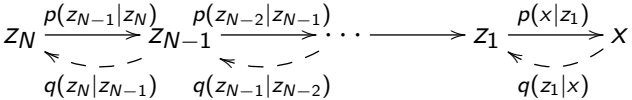
Hierarchical latent variable models

$$z_N \xrightarrow{p(z_{N-1}|z_N)} z_{N-1} \xrightarrow{p(z_{N-2}|z_{N-1})} \cdots \longrightarrow z_1 \xrightarrow{p(x|z_1)} x$$

We want to maximise

$$\log p(x) = \log \int p(z_N) p(z_{N-1} \mid z_N) \cdots p(z_1 \mid z_2)\, p(x \mid z_1)\, dz_1 \cdots dz_N$$

Hierarchical latent variable models

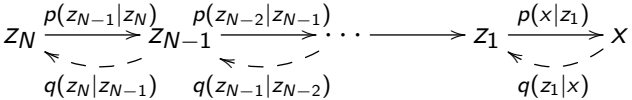


We want to maximise

$$\log p(x) = \log \int p(z_N) p(z_{N-1} \mid z_N) \cdots p(z_1 \mid z_2)\, p(x \mid z_1)\, dz_1 \cdots dz_N$$

Introduce approximate posteriors $q(z_n \mid z_{n-1})$;

Hierarchical latent variable models



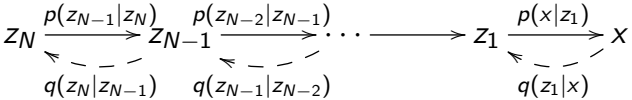
We want to maximise

$$\log p(x) = \log \int p(z_N)p(z_{N-1} \mid z_N) \cdots p(z_1 \mid z_2)\, p(x \mid z_1)\, dz_1 \cdots dz_N$$

Introduce approximate posteriors $q(z_n \mid z_{n-1})$; hierarchical ELBO:

$$\log p(x) \geq \mathbb{E}_{z_1,\ldots,z_N \sim q(\cdots|x)} \left[\log \frac{p(z_N)p(z_{N-1} \mid z_N) \cdots p(x \mid z_1)}{q(z_1 \mid x) \ldots q(z_N \mid z_{N-1})} \right]$$

Hierarchical latent variable models



We want to maximise

$$\log p(x) = \log \int p(z_N)p(z_{N-1} \mid z_N) \cdots p(z_1 \mid z_2)\, p(x \mid z_1)\, dz_1 \cdots dz_N$$

Introduce approximate posteriors $q(z_n \mid z_{n-1})$; hierarchical ELBO:

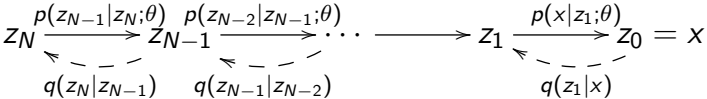
$$\log p(x) \geq \mathbb{E}_{z_1,\ldots,z_N \sim q(\cdots|x)} \left[\log \frac{p(z_N)p(z_{N-1} \mid z_N) \cdots p(x \mid z_1)}{q(z_1 \mid x) \ldots q(z_N \mid z_{N-1})} \right]$$

To train the model given fixed q:

- Sample the latents from q in reverse order: $x \to z_1 \to \cdots \to z_N$
- Maximise $\log p$ in ancestral order w.r.t. parameters of p

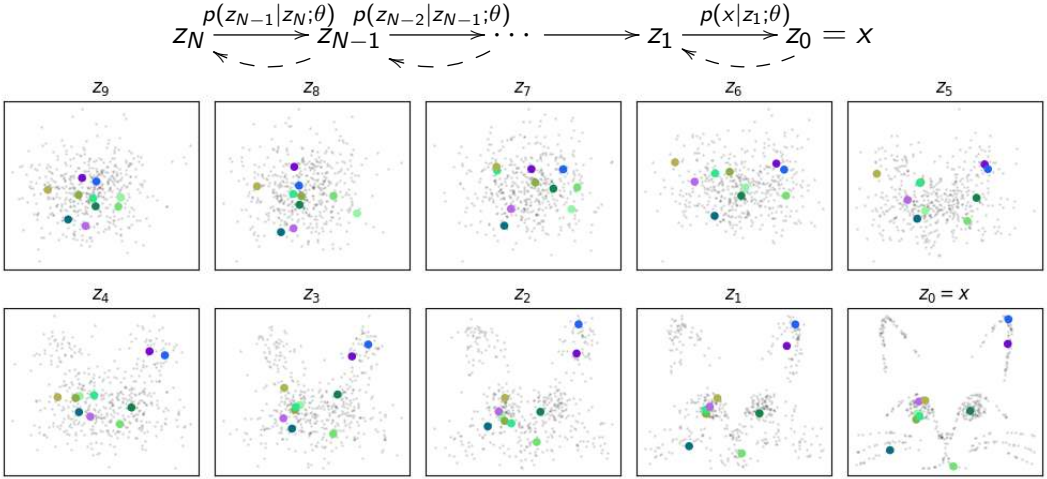
- Historical overview
- Review of latent variable models
- Diffusion models as hierarchical generative models
  - Evaluation and likelihood estimation
  - Latent variable model perspective

Diffusion models are deep VAEs with fixed encoder

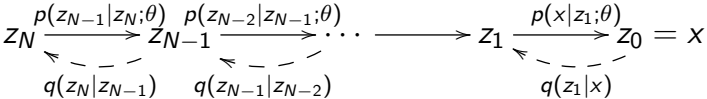


- Diffusion models (typically) fix the distributions $q(z_n \mid z_{n-1})$
  - Typically to adding isotropic Gaussian noise: $q(z_n \mid z_{n-1}) = \mathcal{N}(z_n; z_{n-1}, \sigma_n^2 I_d)$. . . or variants (next time)

Diffusion models are deep VAEs with fixed encoder



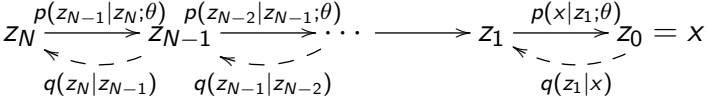
Diffusion models are deep VAEs with fixed encoder



- Diffusion models (typically) fix the distributions $q(z_n \mid z_{n-1})$
  - Typically to adding isotropic Gaussian noise: $q(z_n \mid z_{n-1}) = \mathcal{N}(z_n; z_{n-1}, \sigma_n^2 I_d)$. . . or variants (next time)
- Training to maximise $\log p(x^t)$ for a data sample x^t:
  - Sample the latents from q: $x = z_0 \rightsquigarrow z_1 \rightsquigarrow \cdots \rightsquigarrow z_N$
  - Gradient step on the likelihood in ancestral factorisation:

$$\log [p(z_N; \theta) p(z_{N-1} \mid z_N; \theta) \ldots p(x \mid z_1; \theta)]$$

Diffusion models are deep VAEs with fixed encoder

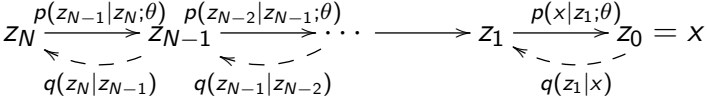


- Diffusion models (typically) fix the distributions $q(z_n \mid z_{n-1})$
  - Typically to adding isotropic Gaussian noise: $q(z_n \mid z_{n-1}) = \mathcal{N}(z_n; z_{n-1}, \sigma_n^2 I_d)$. . . or variants (next time)
- Training to maximise $\log p(x^t)$ for a data sample x^t:
  - Sample the latents from q: $x = z_0 \rightsquigarrow z_1 \rightsquigarrow \cdots \rightsquigarrow z_N$
  - Gradient step on the likelihood in ancestral factorisation:

$$\log [p(z_N; \theta) p(z_{N-1} \mid z_N; \theta) \dots p(x \mid z_1; \theta)]$$

- Typical assumptions:
  - $p(z_{n-1} \mid z_n; \theta)$ is Gaussian with mean computed by a NN with input n, z_n and parameters θ
  - Its variance is fixed to something related to σ_n^2 or σ_{n-1}^2

Diffusion models are deep VAEs with fixed encoder



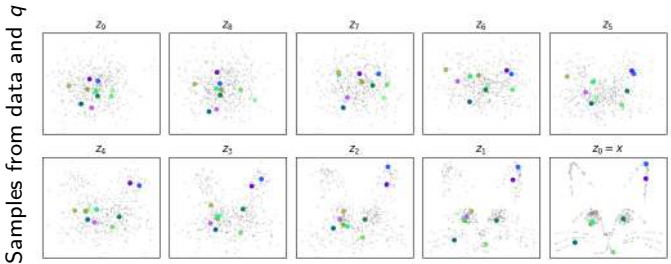
- Diffusion models (typically) fix the distributions $q(z_n \mid z_{n-1})$
  - Typically to adding isotropic Gaussian noise: $q(z_n \mid z_{n-1}) = \mathcal{N}(z_n; z_{n-1}, \sigma_n^2 I_d)$. . . or variants (next time)
- Training to maximise $\log p(x^t)$ for a data sample x^t:
  - Sample the latents from q: $x = z_0 \rightsquigarrow z_1 \rightsquigarrow \cdots \rightsquigarrow z_N$
  - Gradient step on the likelihood in ancestral factorisation:

$$\log [p(z_N; \theta) p(z_{N-1} \mid z_N; \theta) \dots p(x \mid z_1; \theta)]$$

- Typical assumptions:
  - $p(z_{n-1} \mid z_n; \theta)$ is Gaussian (but not always!) with mean computed by a NN with input n, z_n and parameters θ
  - Its variance is fixed to something related to σ_n^2 or σ_{n-1}^2 (but not always!)

Training a diffusion model on 2D data

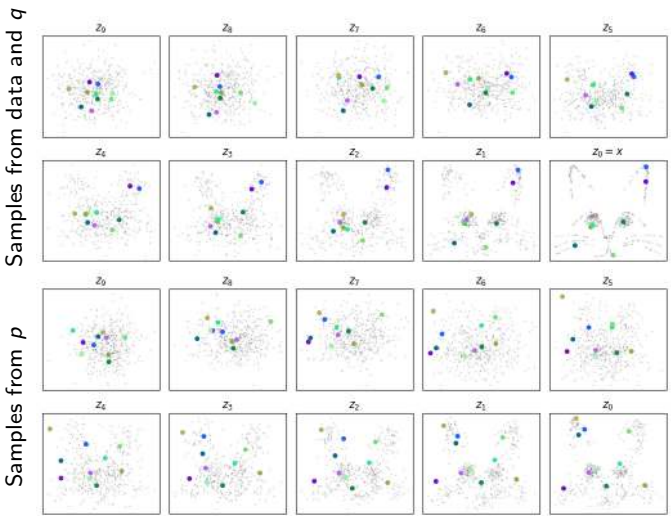
Posterior model
(noising)



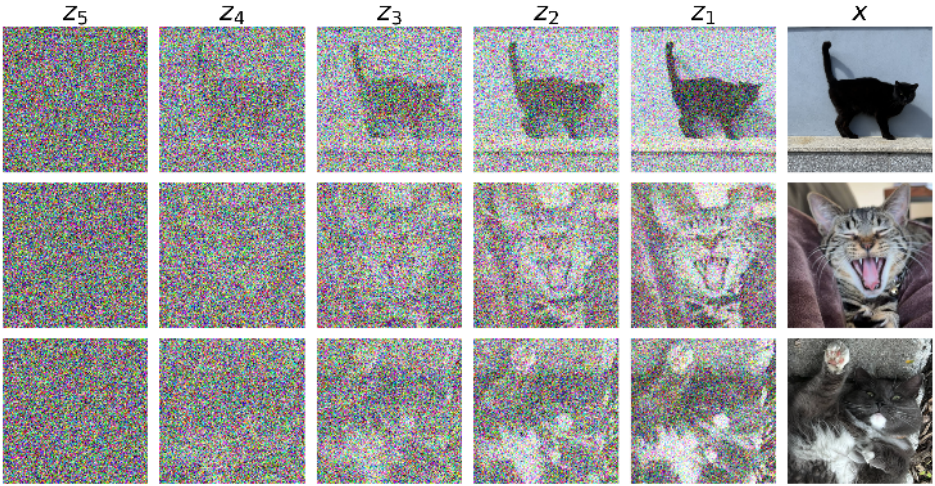
Training a diffusion model on 2D data

Posterior model (noising)

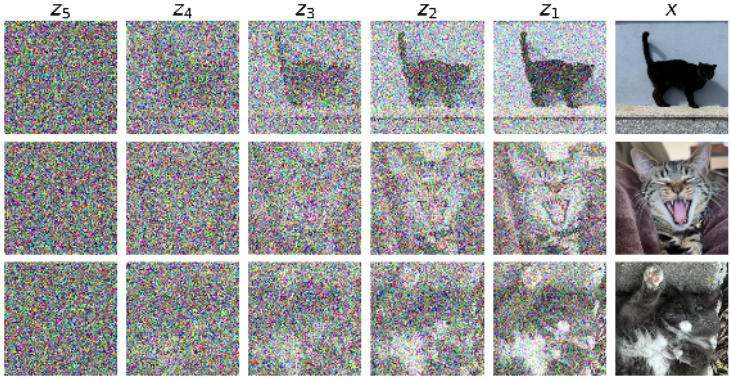
Generative model (denoising / reconstruction)



Scaling to images



Scaling to images



Generating cats with Gaussian $p(z_{n-1} \mid z_n)$ requires **hundreds** of steps

- But much fewer if the p distributions are more expressive
  e.g., [Xiao et al., 'Tackling the generative learning trilemma...']: train the p distributions as GANs!

First observations

- Noise distribution $p(z_N)$ not usually trained (approximately Gaussian)
  - The choice of noise schedule (σ_n) and number of steps are important (more later!)

First observations

- Noise distribution $p(z_N)$ not usually trained (approximately Gaussian)
  - The choice of noise schedule (σ_n) and number of steps are important (more later!)
- To maximise

$$\log\left[p(z_N;\theta)p(z_{N-1} \mid z_N;\theta)\dots p(x \mid z_1;\theta)\right],$$

can randomly sample **one** of the $p(z_{n-1} \mid z_n;\theta)$ for training

First observations

- Noise distribution $p(z_N)$ not usually trained (approximately Gaussian)
  - The choice of noise schedule (σ_n) and number of steps are important (more later!)
- To maximise

$$\log\left[p(z_N;\theta)p(z_{N-1} \mid z_N;\theta)\dots p(x \mid z_1;\theta)\right],$$

  can randomly sample **one** of the $p(z_{n-1} \mid z_n;\theta)$ for training
- Two magic properties to make training efficient:
  - **Simulation-free training:** We can get z_{n-1}, z_n from $z_0 = x$ without all the intermediate steps (why?)

First observations

- Noise distribution $p(z_N)$ not usually trained (approximately Gaussian)
  - The choice of noise schedule (σ_n) and number of steps are important (more later!)
- To maximise

$$\log\left[p(z_N;\theta)p(z_{N-1} \mid z_N;\theta)\dots p(x \mid z_1;\theta)\right],$$

  can randomly sample **one** of the $p(z_{n-1} \mid z_n;\theta)$ for training
- Two magic properties to make training efficient:
  - **Simulation-free training:** We can get z_{n-1}, z_n from $z_0 = x$ without all the intermediate steps (why?)
  - **Rao-Blackwellised denoising objective:** We can estimate the gradient using just z_0 and z_n (next time!)

First observations

- Noise distribution $p(z_N)$ not usually trained (approximately Gaussian)
  - The choice of noise schedule (σ_n) and number of steps are important (more later!)
- To maximise

$$\log\left[p(z_N;\theta)p(z_{N-1} \mid z_N;\theta)\dots p(x \mid z_1;\theta)\right],$$

  can randomly sample **one** of the $p(z_{n-1} \mid z_n;\theta)$ for training
- Two magic properties to make training efficient:
  - **Simulation-free training:** We can get z_{n-1}, z_n from $z_0 = x$ without all the intermediate steps (why?)
  - **Rao-Blackwellised denoising objective:** We can estimate the gradient using just z_0 and z_n (next time!)
  - Generalisations may lose one or both magic properties

Evaluation and tradeoffs

- Evaluation: log-likelihood using ELBO, or sample-based metrics

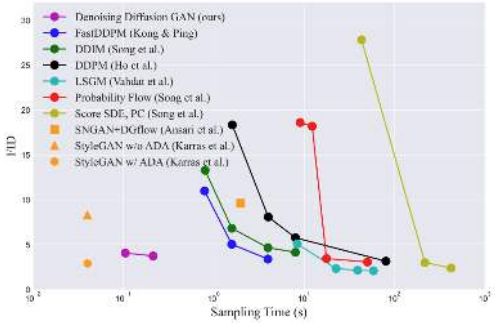
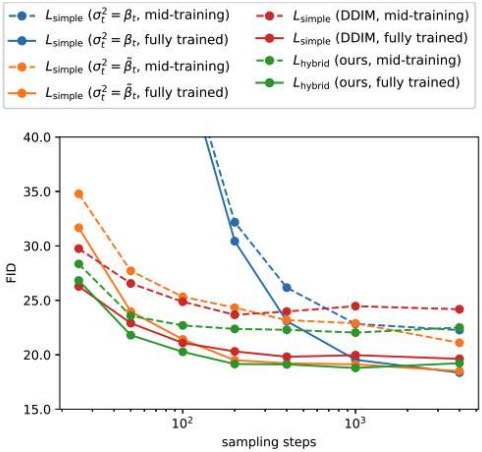


Figure 4: Sample quality vs sampling time trade-off.

[Xiao et al., 'Tackling the generative learning trilemma. . . ']

Evaluation and tradeoffs

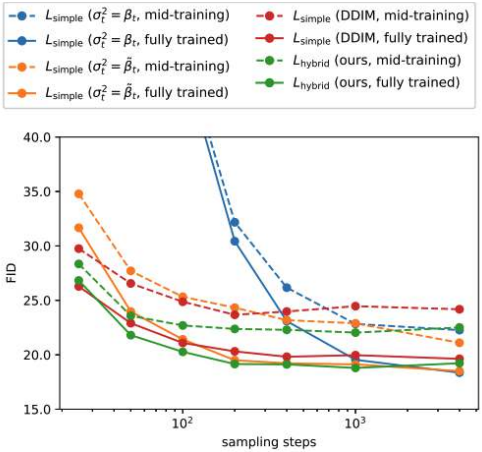
- Evaluation: log-likelihood using ELBO, or sample-based metrics
- Several tradeoffs in modeling choices:
  - **Number of steps:** More steps $\rightsquigarrow$ better samples, but slower training (but more in two weeks!) and sampling



[Nichol&Dhariwal, 'Improved denoising...']

Evaluation and tradeoffs

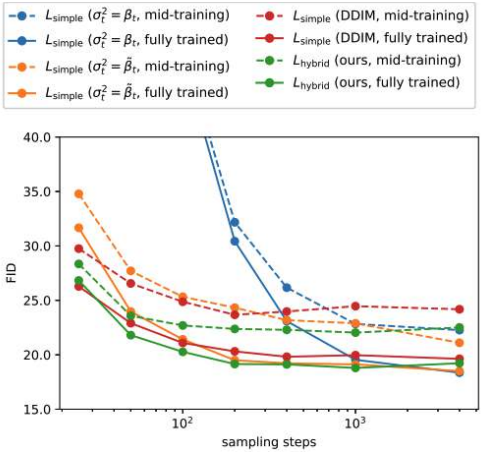
- Evaluation: log-likelihood using ELBO, or sample-based metrics
- Several tradeoffs in modeling choices:
  - **Number of steps:** More steps $\rightsquigarrow$ better samples, but slower training (but more in two weeks!) and sampling
  - **Noise schedule:** Optimal rate σ_n to destroy the signal?



[Nichol&Dhariwal, 'Improved denoising...']

Evaluation and tradeoffs

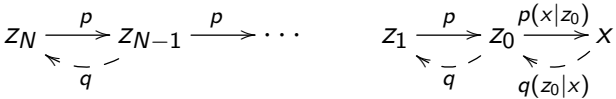
- Evaluation: log-likelihood using ELBO, or sample-based metrics
- Several tradeoffs in modeling choices:
  - **Number of steps:** More steps $\rightsquigarrow$ better samples, but slower training (but more in two weeks!) and sampling
  - **Noise schedule:** Optimal rate σ_n to destroy the signal?
  - **Model complexity:** More expressive p $\rightsquigarrow$ better samples, harder to train



[Nichol&Dhariwal, 'Improved denoising...']

Advantages of the LVM perspective

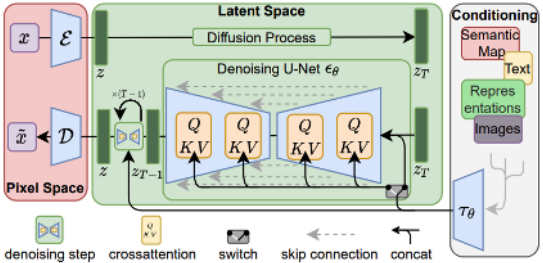
- **Latent diffusion:** separate learnt encoder and decoder for $z_0 \leftrightarrow x$ (most large-scale image models work this way)



Advantages of the LVM perspective

- **Latent diffusion:** separate learnt encoder and decoder for $z_0 \leftrightarrow x$ (most large-scale image models work this way)

$$z_N \underset{q}{\overset{p}{\rightleftarrows}} z_{N-1} \underset{q}{\overset{p}{\rightleftarrows}} \cdots \qquad z_1 \underset{q}{\overset{p}{\rightleftarrows}} z_0 \underset{q(z_0|x)}{\overset{p(x|z_0)}{\rightleftarrows}} x$$



[Rombach et al., 'High-resolution image synthesis with latent diffusion models']

Advantages of the LVM perspective

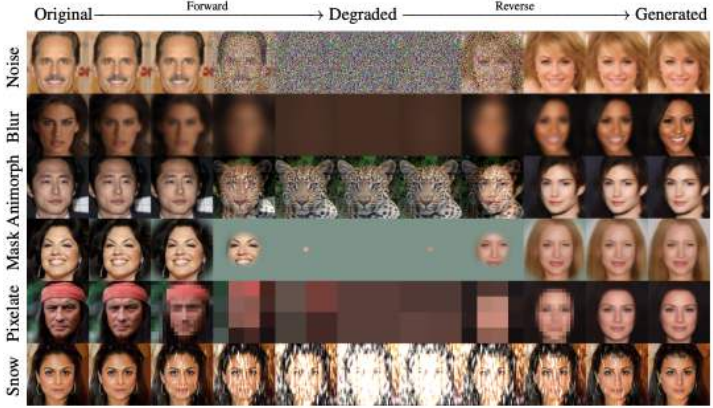
- **Latent diffusion:** separate learnt encoder and decoder for $z_0 \leftrightarrow x$ (most large-scale image models work this way)

Advantages of the LVM perspective

- **Latent diffusion:** separate learnt encoder and decoder for $z_0 \leftrightarrow x$ (most large-scale image models work this way)
- **Non-Gaussian noising:** Define the q by other means than Gaussian noise

Advantages of the LVM perspective

- **Latent diffusion:** separate learnt encoder and decoder for $z_0 \leftrightarrow x$ (most large-scale image models work this way)
- **Non-Gaussian noising:** Define the q by other means than Gaussian noise



[Bansal et al., 'Cold diffusion. . . ']

Conclusion and looking forward

Next time:

- The 'second magic property' and its connection to **denoising score matching**
- Mechanics of linear drift-diffusion noising processes
  - The magic properties + the right noising process are enough to train a diffusion model on moderately-sized image data

Conclusion and looking forward

Next time:

- The 'second magic property' and its connection to **denoising score matching**
- Mechanics of linear drift-diffusion noising processes
  - The magic properties + the right noising process are enough to train a diffusion model on moderately-sized image data
- Conditioning and guidance

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there wa
some unprocessed data that should have been added to the final page this
extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page will go aw
because LaTeX now knows how many pages to expect for this document.