Deep generative modelling

Concepts and characteristics

Chris G. Willcocks

Durham University

About

Background

- Associate professor at Durham university computer science (north-east England)
  - Beautiful historic cathedral city
- Research in deep generative modelling
- Lecturer of deep learning, reinforcement learning and cyber security
- Family in Sha Tin Wai :)



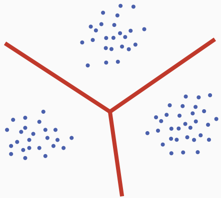


Why generative modelling?

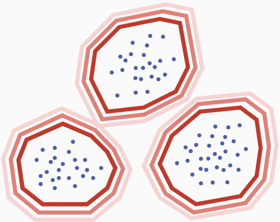
Why generative modelling?

- Likelihood (density) estimates: how close model is to true distribution
- Uncertainty quantification
- Can convert more easily to discriminative model
- Sampling
- The beating heart of AI advances!

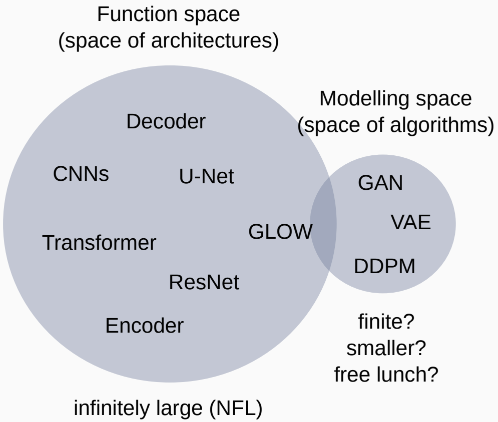
Discriminative modelling
(classification, regression)



Generative modelling
(sampling, density estimation)



The modelling space (not the function space)



What is the modelling space?

- We mean the space of modelling approaches
- That's not to say the function space is not important...
- But we have no-free-lunch theorem

Modelling approaches (the key equations)

Summary

There are several key approaches that we will cover today:

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i | x_1, ..., x_{i-1})$$

$$p(\mathbf{x}) \approx \frac{e^{-E(\mathbf{x})}}{\int_{\tilde{\mathbf{x}} \in \mathcal{X}} e^{-E(\tilde{\mathbf{x}})}} \quad \text{e.g. } \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

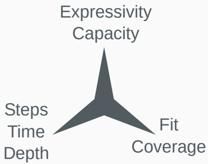
$$\log p(\mathbf{x}) \geq \mathcal{L}_{\text{recon}}^{\text{pixel}} - D_{\mathsf{KL}} \left[q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}) \right]$$

$$\log p(\mathbf{x}) \neq \log D(\mathbf{x}) \quad \text{(in GAN)}$$

$$p(\mathbf{x}) = p_Z(f^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

$$p_\theta(x) \approx p(x) \text{ when } \min_\theta \mathsf{OT}(G_\theta(z), x)$$

Method	Train Speed	Sample Speed	Num. Params.	Resolution Scaling	Free-form Jacobian	Exact Density	FID	NLL (in BPD)
Generative Adversarial Networks								
DCGAN [182]	★★★★★	★★★★★	★★★★★	★★★☆☆	✓	✗	37.11	-
ProGAN [114]	★★★★☆	★★★★★	★★★★☆	★★★★★	✓	✗	15.52	-
BigGAN [19]	★★★★☆	★★★★★	★★★★★	★★★★☆	✓	✗	14.73	-
StyleGAN2 + ADA [115]	★☆☆☆☆	★★★★★	★★★☆☆	★★★★★	✓	✗	2.42	-
Energy Based Models								
IGEBM [46]	★★★☆☆	★★☆☆☆	★★★★★	★★★☆☆	✓	✗	37.9	-
Denoising Diffusion [87]	★★★★★	★★☆☆☆	★★★☆☆	★★★★☆	✓	(✓)	3.17	$\leq$ 3.75
DDPM++ Continuous [206]	★★★★★	★★☆☆☆	★★★☆☆	★★★★☆	✓	(✓)	2.20	-
Flow Contrastive (EBM) [55]	★☆☆☆☆	★★☆☆☆	★★☆☆☆	★★☆☆☆	✓	✗	37.30	$\approx$ 3.27
VAEBM [247]	★★☆☆☆	★★★★☆	★☆☆☆☆	★★★★☆	✓	✗	12.19	-
Variational Autoencoders								
Convolutional VAE [123]	★★★★★	★★★★★	★★★★★	★★☆☆☆	✓	(✓)	106.37	$\leq$ 4.54
Variational Lossy AE [29]	★★★☆☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆	✗	(✓)	-	$\leq$ 2.95
VQ-VAE [184], [235]	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★★★	✗	(✓)	-	$\leq$ 4.67
VD-VAE [31]	★☆☆☆☆	★★★★★	★★★☆☆	★★★★★	✓	(✓)	-	$\leq$ 2.87
Autoregressive Models								
PixelRNN [234]	★★☆☆☆	★☆☆☆☆	★☆☆☆☆	★★★☆☆	✗	✓	-	3.00
Gated PixelCNN [233]	★★★☆☆	★☆☆☆☆	★☆☆☆☆	★★★☆☆	✗	✓	65.93	3.03
PixelIQN [173]	★★★☆☆	★☆☆☆☆	★☆☆☆☆	★★★☆☆	✗	✓	49.46	-
Sparse Trans. + DistAug [32], [110]	★★★★☆	★☆☆☆☆	★★★☆☆	★★★☆☆	✗	✓	14.74	2.66
Normalizing Flows								
RealNVP [43]	★★☆☆☆	★★★★★	★★★★★	★★★☆☆	✗	✓	-	3.49
GLOW [124]	★☆☆☆☆	★★★★★	★★☆☆☆	★★★★☆	✗	✓	45.99	3.35
FFJORD [62]	★☆☆☆☆	★★★☆☆	★★★★★	★★☆☆☆	✓	(✓)	-	3.40
Residual Flow [26]	★★☆☆☆	★★★★☆	★★★★☆	★★★★☆	✓	(✓)	46.37	3.28



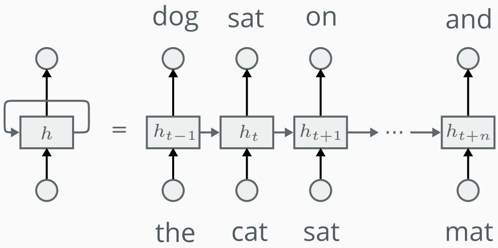
Autoregressive generative models

Definition: autoregressive (AR) generative models (e.g. chatGPT)

AR models maximise the likelihood of the training data (excellent mode coverage):

$$p_\theta(\mathbf{x}) = p_\theta(x_1, ..., x_N) = \prod_{i=1}^{N} p_\theta(x_i | x_1, ..., x_{i-1})$$

This is slow due to the sequential nature defined by the chain rule of probability.



Diffusion probablistic models (DPMs)

Definition: diffusion probablistic models (DPMs)

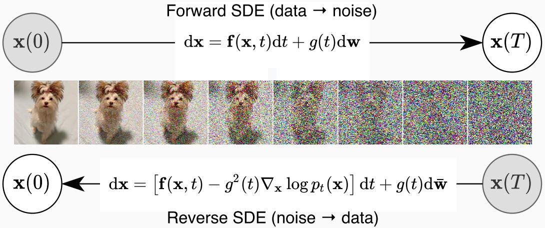
These similarly define a forward (diffusion) equation:

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

which can be reversed to sample the model:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

This also requires a long iterative **transformation process**.



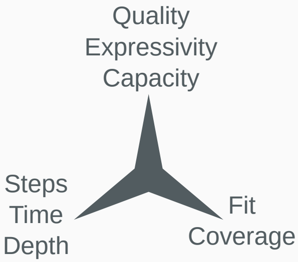
Reflection

Question: where are these in the trilemma?

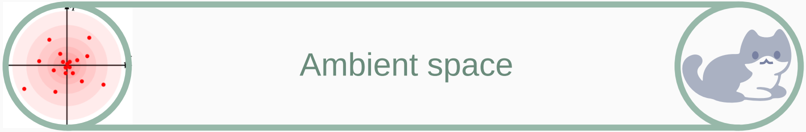
- They are slow (lots of iterations)
- They have excellent coverage
- They have excellent quality

...let's try and identify some high-level concepts & characteristics.

The generative modelling trilemma
(empirical observation)



Is the loss in the output space or the ambient space?



Input space
(state, prior)

Output space
(state', data)

How about VAEs?

Variational autoencoders (VAEs)

Definition: variational autoencoders

VAEs have a **reconstruction loss on the output** and a latent loss:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right] = \mathcal{L}_{\text{recon}}^{\text{pixel}} + \mathcal{L}_{\text{prior}}$$

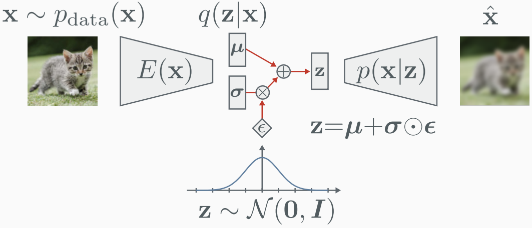
where

$$\mathcal{L}_{\text{recon}}^{\text{pixel}} = -\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})],$$
$$\mathcal{L}_{\text{prior}} = D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})).$$

They're trained with the reparameterisation trick to maximise the evidence lower bound (ELBO):

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\text{KL}}\left[q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})\right]$$

... must be balanced and just a bound; they suck!

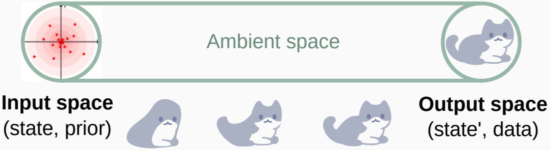


Reflection (why can progressive approaches work well?)

"Very Deep VAEs Generalize Autoregressive Models"

—much better modelling quality, as has multiple losses in the ambient space. But slower!

Low resolution → High resolution



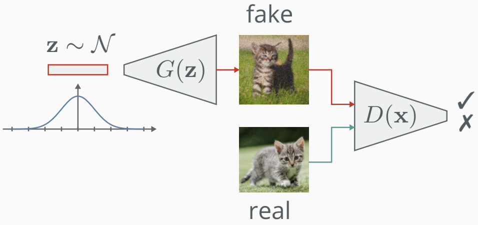
How about GANs?

Definition: generative adversarial networks

A generative adversarial network (GAN) is a non-coorporative zero-sum game where two networks compete against each other [1].

One network $G(\mathbf{z})$ generates new samples, whereas D estimates the probability the sample was from the training data rather than G:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))].$$



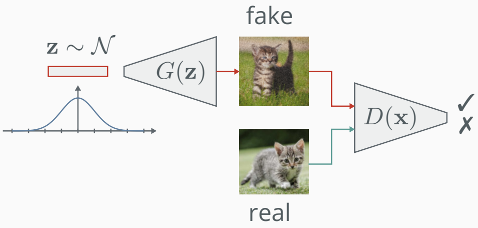
Generative adversarial networks properties

GAN properties

GANs benefit from differentiable data augmentation for both reals and fakes, but are otherwise notoriously difficult to train:

- Non-convergence
- Diminishing gradient
- Difficult to balance
- Mode collapse (next slide)

Link to Colab example



Definition: mode collapse

This is where the generator rotates through a small subset of outputs, and the discriminator is unable to get out of the trap. Mode collapse is arguably the main limitation of GANs.

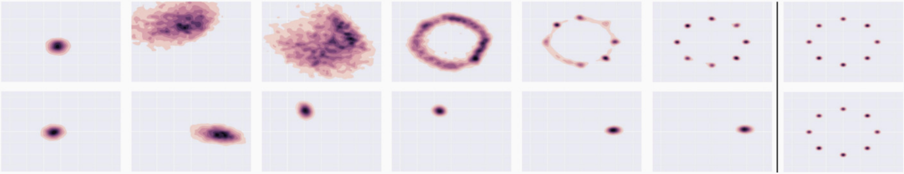
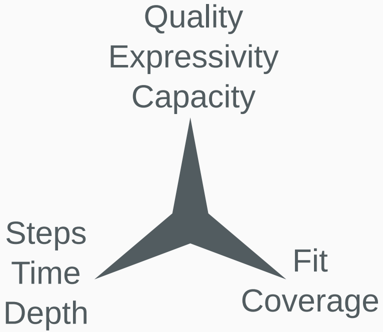


Figure from "Unrolled generative adversarial networks". The final column shows the target data distribution and the bottom row shows a GAN rotating through the modes.

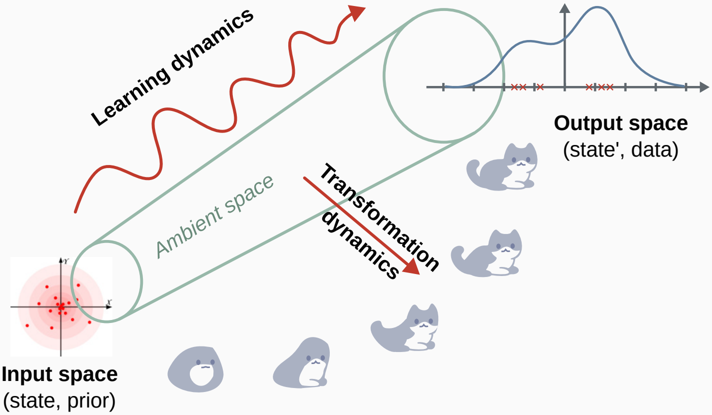
Reflection (where are GANs in the trilemma?)

The generative modelling trilemma
(empirical observation)



Reflection (two temporal dimensions)

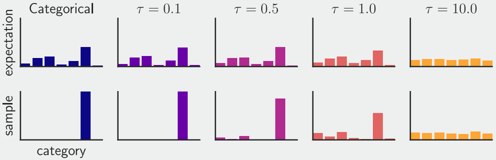
Is the path more curved in the learning or transofmration dynamics?



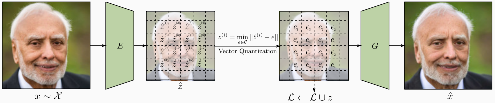
Intentional mode collapse: vector quantization

Vector quanitization

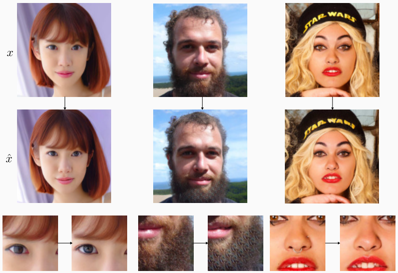
Imposing a discrete prior on the latents can be achieved with either variational or adversarial (non-blurry) approaches.



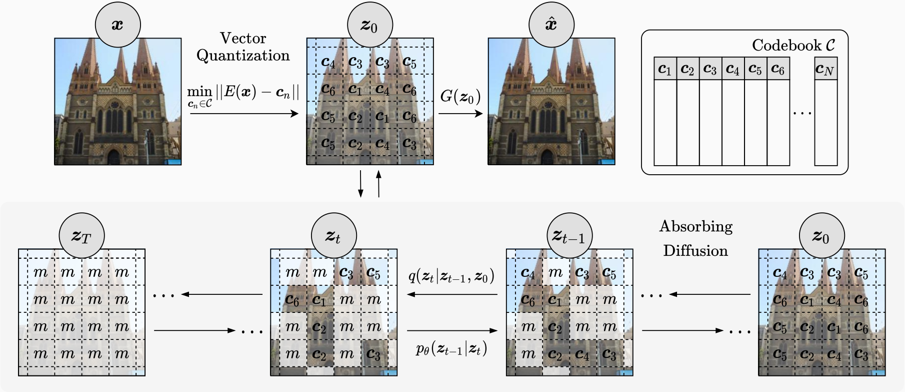
The Gumbel-Softmax distribution interpolates between discrete one-hot-encoded categorical distributions and continuous categorical densities.



Above: vector quantisation. **Below:** shift mode collapse to perceptually unimportant parts of the signal.



Intentional mode-collapse with diverse, globally-coherent absorbing diffusion.



Link to project page

Figure from Unleashing Transformers, ECCV2022, Bond-Taylor, Willcocks et. al [2].

Our hybrid [3] 2 seconds generation, 2 days training, single GTX 1080Ti

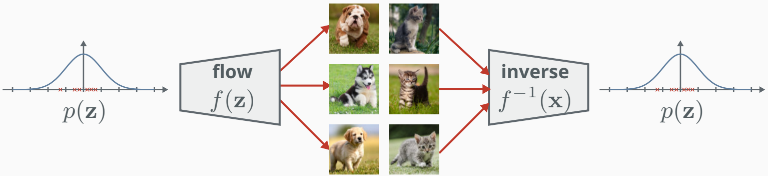
Reflection

How about normalising flows?

Flow models definition

Definition: flow models

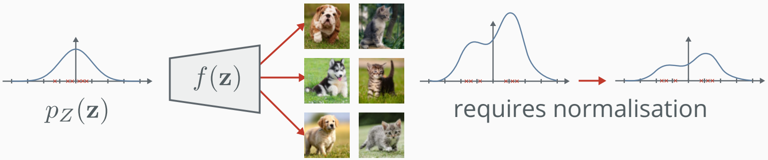
Flow models restrict our function to be a chain of invertible functions, called a flow, therefore the whole function is invertible.



Definition: the change of variables theorem

Given $p_Z(\mathbf{z})$ where $\mathbf{x} = f(\mathbf{z})$ and $\mathbf{z} = f^{-1}(\mathbf{x})$ we ask what is $p_X(\mathbf{x})$?

$$p_X(\mathbf{x}) = p_Z(f^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

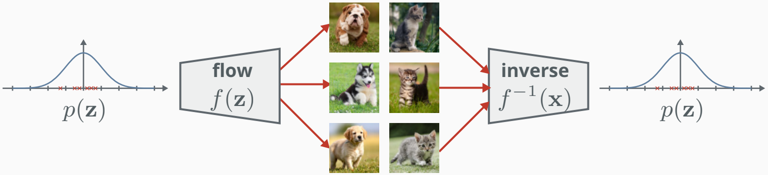


Definition: normalising flows

Normalising flows $f : \mathbb{R}^n \to \mathbb{R}^n$ transform and renormalise a sample $\mathbf{z} \sim p_\theta(\mathbf{z})$ through a chain of bijective transformations f, where:

$$\mathbf{x} = f_\theta(\mathbf{z}) = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z})$$

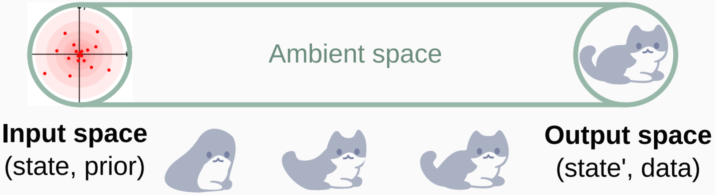
$$\log p_\theta(\mathbf{x}) = \log p_\theta(\mathbf{z}) + \sum_{i=1}^{K} \log \left| \det \left(\frac{\partial f_i^{-1}}{\partial \mathbf{z}_i} \right) \right|$$

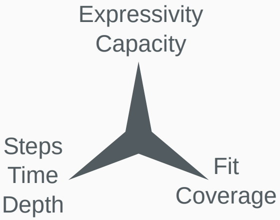


Normalising flows normalising flow layers

Description	Function	Log-Determinant
Additive Coupling	$\mathbf{y}^{(1:d)} = \mathbf{x}^{(1:d)}$ $\mathbf{y}^{(d+1:D)} = \mathbf{x}^{(d+1:D)} + f(\mathbf{x}^{(1:d)})$	0
Planar	$\mathbf{y} = \mathbf{x} + \mathbf{u}h(\mathbf{w}^T\mathbf{z} + b)$ With $\mathbf{w} \in \mathbb{R}^D$, $\mathbf{u} \in \mathbb{R}^D$, $\mathbf{b} \in \mathbb{R}$	$\ln\lvert 1 + \mathbf{u}^T h'(\mathbf{w}^T\mathbf{z} + b)\mathbf{w}\rvert$
Affine Coupling	$\mathbf{y}^{(1:d)} = \mathbf{x}^{(1:d)}$ $\mathbf{y}^{(d+1:D)} = \mathbf{x}^{(d+1:D)} \odot f_\sigma(\mathbf{x}^{(1:d)}) + f_\mu(\mathbf{x}^{(1:d)})$	$\sum_1^d \ln\lvert f_\sigma(x^{(i)})\rvert$
Batch Normalization	$\mathbf{y} = \frac{\mathbf{x} - \tilde{\mu}}{\sqrt{\tilde{\sigma}^2 + \epsilon}}$	$-\frac{1}{2}\sum_i \ln\left(\tilde{\sigma}_i^2 + \epsilon\right)$
1x1 Convolution [4]	With $h \times w \times c$ tensor $\mathbf{x}$ & $c \times c$ tensor $\mathbf{W}$ $\forall i, j : \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$h \cdot w \cdot \ln\lvert\det \mathbf{W}\rvert$
i-ResNet	$\mathbf{y} = \mathbf{x} + f(\mathbf{x})$ where $\lVert f\rVert_L < 1$	$\mathrm{tr}(\ln(\mathbf{I} + \nabla_\mathbf{x} f)) =$ $\sum_{k=1}^{\infty}(-1)^{k+1}\frac{\mathrm{tr}((\nabla_\mathbf{x} f)^k)}{k}$
Emerging Convolutions	$\mathbf{k} = \mathbf{w}_1 \odot \mathbf{m}_1, \qquad \mathbf{g} = \mathbf{w}_2 \odot \mathbf{m}_2$ $\mathbf{y} = \mathbf{k} \star_l (\mathbf{g} \star_l \mathbf{x})$	$\sum_c \ln\lvert\mathbf{k}_{c,c,m_y,m_x}\mathbf{g}_{c,c,m_y,m_x}\rvert$

Reflection (what do normalising flows do?)





—but you need a *lot* of layers, so they kinda suck.

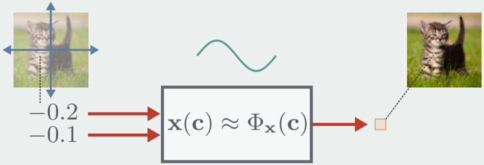
Infinite dimensional models

Definition: SIREN

SInusoidal REpresentation Networks (SIREN) are a simple implicit representation network with fully connected layers, but use `sin` (with clever initialisation to scale it appropriately) as their choice of non-linearity [5].

`sin` is periodic, so it allows to capture patterns over all of the coordinate space (it's translation invariant, like convolutions).

Example: SIREN (implicit network)



Link to project page

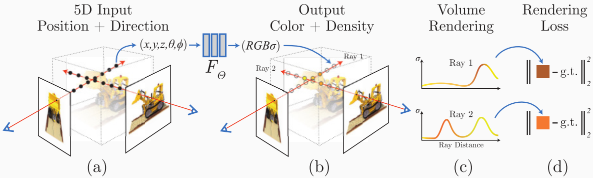
—these are not generative models!

Implicit representation networks NeRF

Definition: NeRF

Neural Radiance Fields (NeRF) are similar to SIRENs, but instead of representing an image, they represent a single 3D scene [6].

They map from pixel positions (x, y, z) and a viewing direction (θ, ϕ) to a colour and density value σ integrated via a ray on F_θ.



Link to project page

—these are not generative models!

Definition: gradient origin networks

Gradient origin networks (GON) treat the derivative of the decoder as an encoder [7]. This allows us to compute the latents:

$$\mathbf{z} = -\nabla_{\mathbf{z}_0} \mathcal{L}(\mathbf{x}, F(\mathbf{z}_0))$$

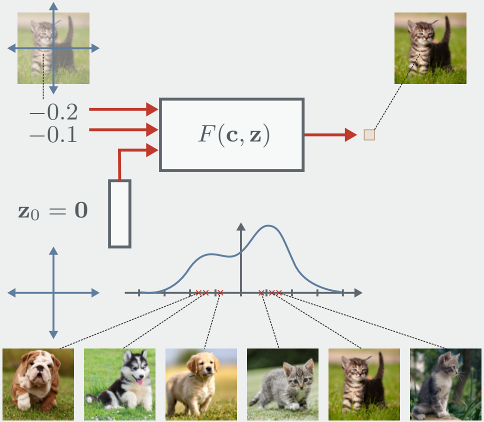
which are jointly optimised, giving GON loss:

$$G = \mathcal{L}(\mathbf{x}, F(-\nabla_{\mathbf{z}_0} \mathcal{L}(\mathbf{x}, F(\mathbf{z}_0)))).$$

Link to project page

—amazingly we found these nearly always outperform autoencoders without encoders
—these can now be generative models

Example: implicit GON



Infinite dimensional diffusion our recent contribution

Definition: ∞-diffusion [8]

We extended diffusion models to infinite dimensions, trained only on a random subset of coordinates, without requiring any compression or discretisation (submitted to NeurIPS 2023).

This is achieved by introducing a mollifier to give some smoothness and locality, a requirement for the neural operators that can operate in the infinite-dimensional Hilbert space.

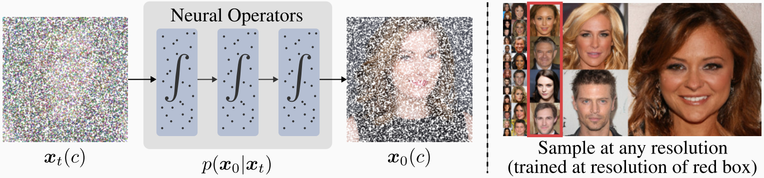


Figure 1: We define a diffusion process in an infinite dimensional image space by randomly sampling coordinates and training a model parameterised by neural operators to denoise at those coordinates.

Lastly, optimal transport in the data space

Optimal transport decoder

Definition: optimal transport decoder

Very few people seem to know you can simply do this (if your dataset is small enough):

$$\mathcal{L}_{\text{OT}} = \mathbb{E}_{x \sim p_d, z \sim p_z} \left[\text{OT}(G_\theta(z), x) \right]$$

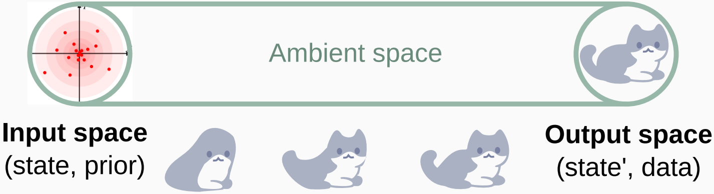
Sinkhorn is a very fast approximation that works well in practice (a little bit slower than an L2 norm).

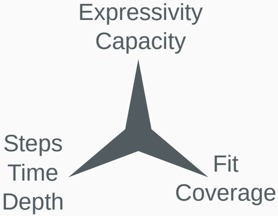
—if the dataset is not small enough, you can still do it and the results are better than a VAE (reminder: VAEs suck!).

Code example: OT decoder

```
xb = next(train_iterator)
xb = xb.to(device)

while (True):

    # xb = next(train_iterator)
    # xb = xb.to(device)

    # forward pass
    z = torch.randn(xb.size(0), args['latent_dim'])
    x_hat = decoder(z)
    loss = sinkhorn_ot_loss(x_hat, xb)

    # update
    opt.zero_grad()
    loss.backward()
    opt.step()
```

Reflection (optimal transport in output space)





—if we could do OT, in few steps, along an infinite-dimensional ambient space, that would be great!

Contributions

- We extended probabilistic diffusion models to **infinite dimensions** without compression ∞-diff [8]
- We showed you **don't always need encoders** (GONs) and generalised implicit networks without hypernetworks [7]
- Building on [9], provided early **theoretical insights** into the trilemma from functional analysis, and considered the modelling space as a coupled system of SDEs with two temporal dimensions.

References I

[1] Ian Goodfellow et al. “Generative adversarial nets”. In: Advances in neural information processing systems. 2014, pp. 2672–2680.

[2] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. “Unleashing Transformers: Parallel Token Prediction with Discrete Absorbing Diffusion for Fast High-Resolution Image Generation from Vector-Quantized Codes”. In: European Conference on Computer Vision (ECCV) (2022).

[3] Alex F McKinney and Chris G Willcocks. “Megapixel Image Generation with Step-Unrolled Denoising Autoencoders”. In: arXiv preprint arXiv:2206.12351 (2022).

[4] Durk P Kingma and Prafulla Dhariwal. “Glow: Generative flow with invertible 1x1 convolutions”. In: Advances in neural information processing systems. 2018, pp. 10215–10224.

[5] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. “Implicit neural representations with periodic activation functions”. In: Advances in Neural Information Processing Systems 33 (2020).

References II

[6] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: European conference on computer vision. Springer. 2020, pp. 405–421.

[7] Sam Bond-Taylor and Chris G. Willcocks. "Gradient Origin Networks". In: International Conference on Learning Representations. 2021. URL: https://dro.dur.ac.uk/34356/1/34356.pdf.

[8] Sam Bond-Taylor and Chris G Willcocks. "∞-Diff: Infinite Resolution Diffusion with Subsampled Mollified States". In: arXiv preprint arXiv:2303.18242 (2023).

[9] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models". In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2021), pp. 1–1. DOI: 10.1109/TPAMI.2021.3116668.