

Deep Generative Models

杨帆

fanyang01@zju.edu.cn

2019年6月20日

Outline

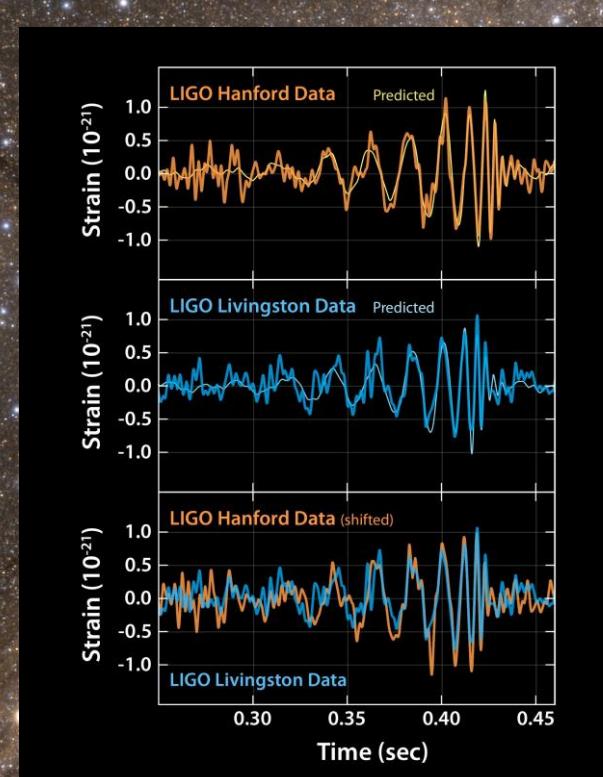
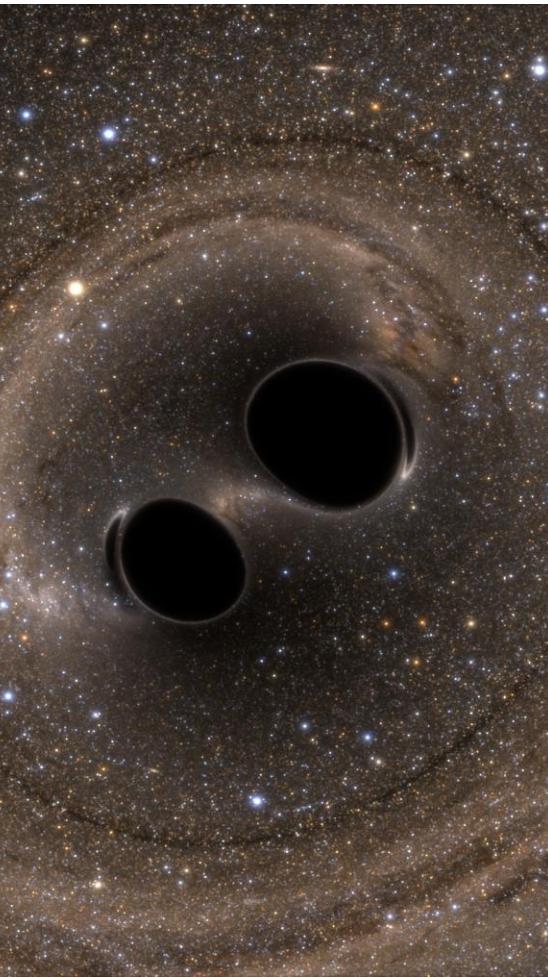
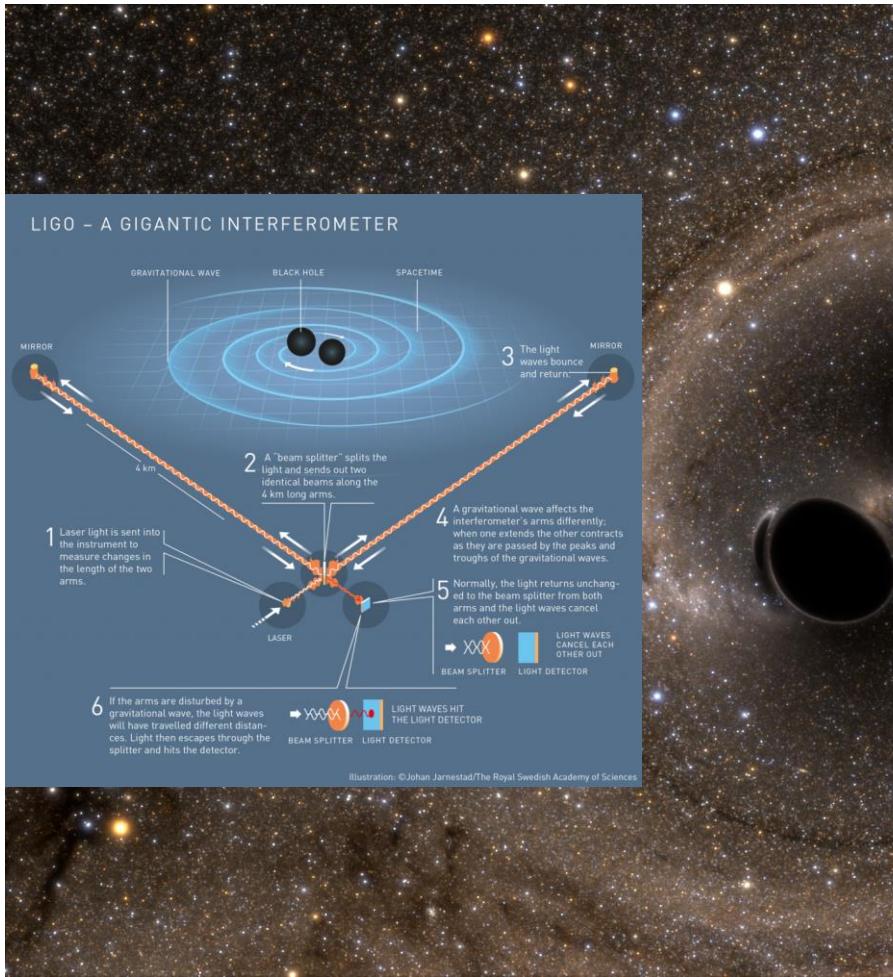
- What, Why
- Landscape of DGMs
- Sequential DGMs
- Summary & References

Outline

- **What, Why**
- Landscape of DGMs
- Sequential DGMs
- Summary & References

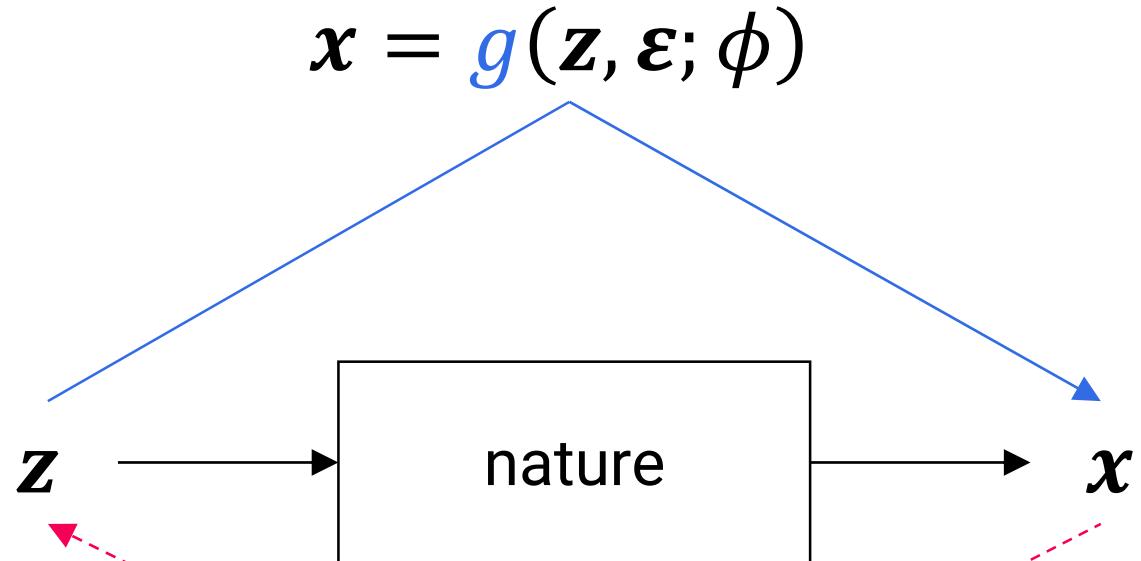
The Nature of Data

- All data were collected by observing some generating processes



Discriminative vs Generative

- Generative model

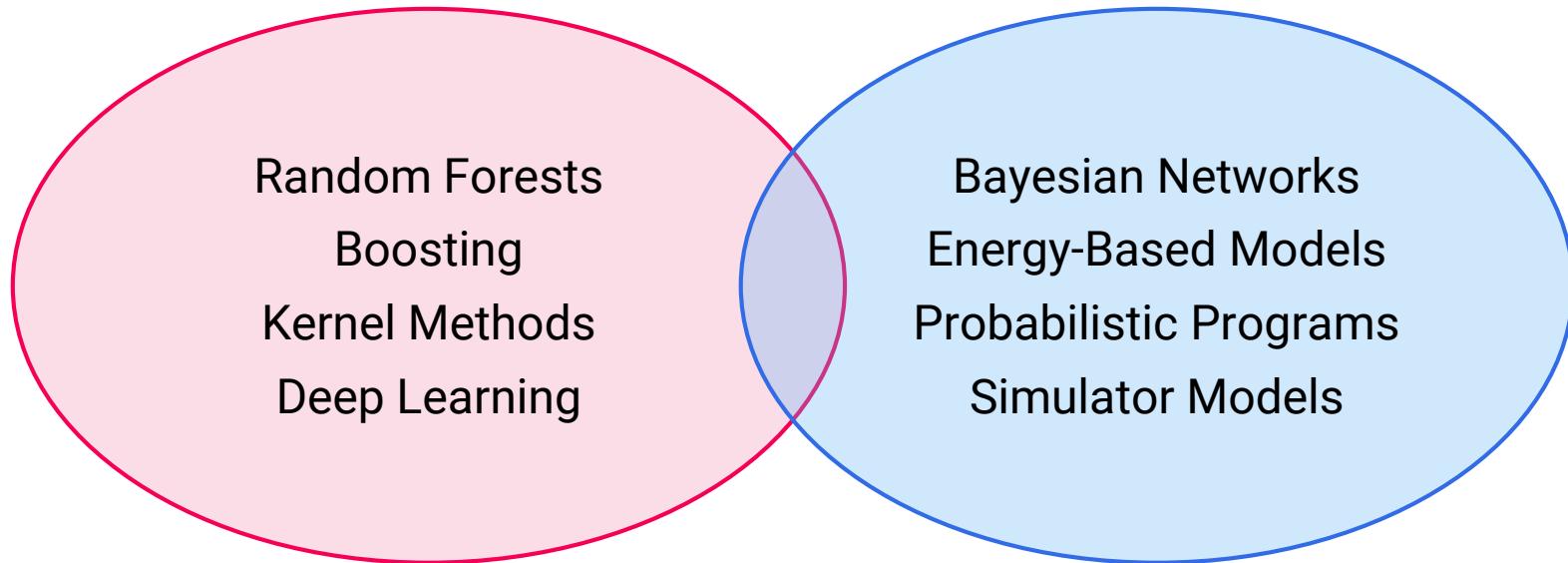


- Data generating process

- Discriminative model

$$z = f(x; \theta) \text{ or } z \sim p(\cdot | x; \theta)$$

Discriminative vs Generative



- Flexible map from input to target
 - Efficient training algorithms
 - Very successful and accurate
 - Good at **inference**
 - ML: the CS version?
 - Alchemy?
- Model causal relations and physics
 - Used by most scientific disciplines
 - Interpretable and data efficient
 - Good at **prediction**
 - ML: the STAT version?
 - True science?

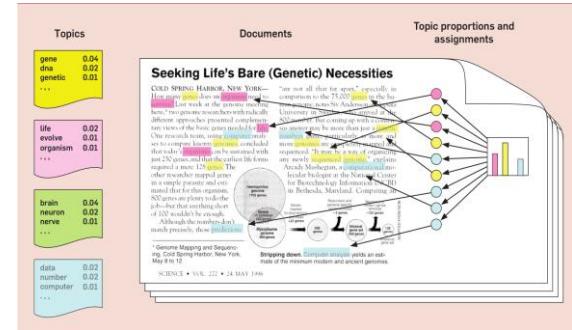
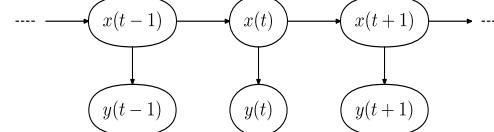
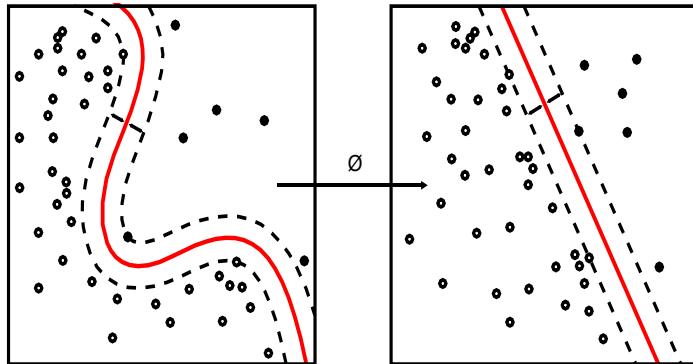
2001

Statistical Modeling: The Two Cultures

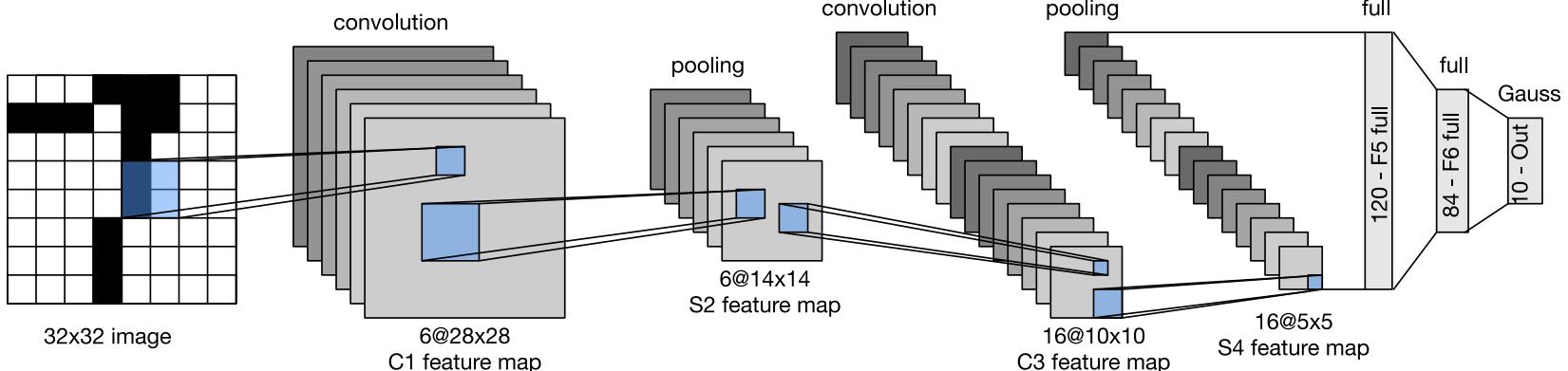
Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

1990 ~ 2010



2010 ~ Now



2014 ~ : The Rise of Probabilistic/Bayesian DL

Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

Abstract

How can we perform efficient inference and learning in directed probabilistic models, in the presence of continuous latent variables with intractable posterior distributions, and large datasets? We introduce a stochastic variational inference and learning algorithm that scales to large datasets and, under some mild differentiability conditions, even works in the intractable case. Our contribution is two-fold. First, we show that a reparameterization of the variational lower bound yields a lower bound estimator that can be straightforwardly optimized using standard stochastic gradient methods. Second, we show that for i.i.d. datasets with continuous latent variables per datapoint, posterior inference can be made tractably efficient by fitting an approximate inference model (also called a recognition model) to the intractable posterior using the proposed lower bound estimator. Theoretical advantages are reflected in experimental results.

Generative Adversarial Nets

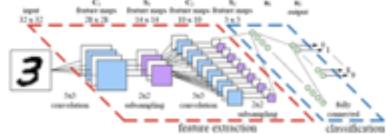
Ian J. Goodfellow*, **Jean Pouget-Abadie†**, **Mehdi Mirza**, **Bing Xu**, **David Warde-Farley**,
Sherjil Ozair‡, **Aaron Courville**, **Yoshua Bengio§**
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D , a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

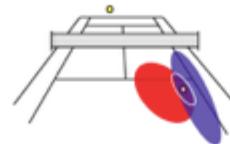
Why should we care?

Two Streams of Machine Learning



Deep Learning

- + Rich non-linear models for classification and sequence prediction.
- + Scalable learning using stochastic approximation and conceptually simple.
- + Easily composable with other gradient-based methods.
- Only point estimates.
- Hard to score models, do selection and complexity penalisation.



Probabilistic Reasoning

- Mainly conjugate and linear models.
- Potentially intractable inference, computationally expensive or long simulation time.
- + Unified framework for model building, inference, prediction and decision making.
- + Explicit accounting for uncertainty and variability of outcomes.
- + Robust to overfitting; tools for model selection and composition.

Complementary strengths, making it natural to combine them

Tutorial on Deep Generative Models. Shakir Mohamed and Danilo Rezende. UAI 2017 Tutorial

Why should we care?

ML as a bag of tricks

Fast special cases:

- K-means
- Kernel Density Estimation
- SVMs
- Boosting
- Random Forests

Extensible family:

- Mixture of Gaussians
- Latent variable models
- Gaussian processes
- Deep neural nets
- Bayesian neural nets

Regularization as bag of tricks

Fast special cases:

- Early stopping
- Ensembling
- L2 Regularization
- Gradient noise
- Dropout

Extensible family:

- Stochastic variational inference

Losses are log-likelihoods

- Squared loss is just unnormalized Normal log-pdf
- “Cross-entropy” now means Categorical log-pmf ?!
- Actual definition: $H(p, q) = - \sum_x p(x) \log q(x)$
- “Teacher forcing” is just evaluating the likelihood of a sequential model $p(x) = \prod_i p_\theta(x_i | x_{<i})$

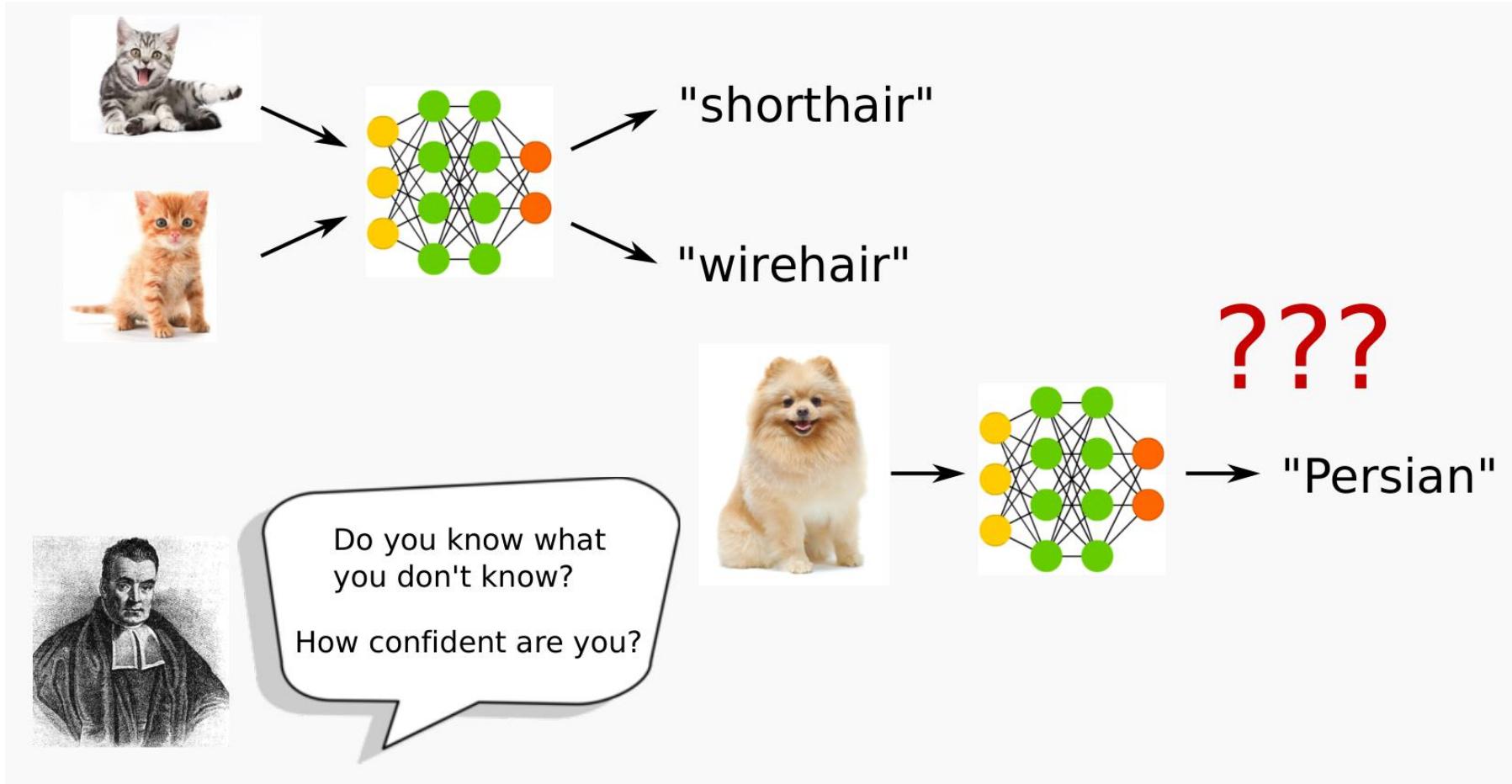
AI as a bag of tricks

Russel and Norvig's parts of AI:

Extensible family:

- Machine learning
- Natural language processing
- Knowledge representation
- Automated reasoning
- Computer vision
- Robotics
- Deep probabilistic latent-variable models + decision theory

Why should we care?



Efficient Computation for Bayesian Deep Learning. Yingzhen Li. 2018

Bayesian
Inference



Monte
Carlo
Sampling



Probabilistic
Graphical
Models



Bayesian
NN



Variational
Inference



Deep
Generative
Models



MLP, CNN,
RNN, GNN

Gradient
Decent,
Back-
propagation

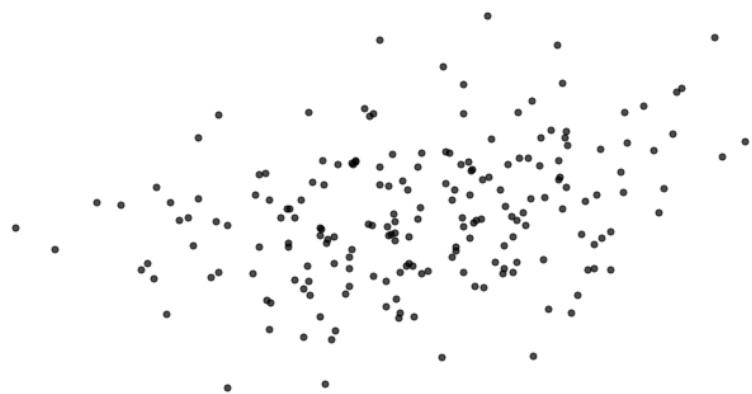
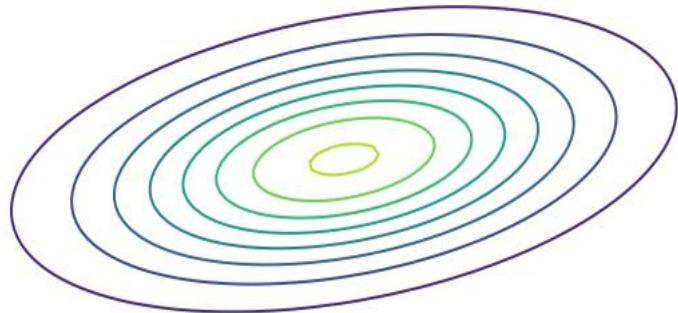
Seq2Seq,
Attention,
ResNet

What is Generative Modeling?

Generative modeling is the art and science of engineering a family of probability distributions that is simultaneously rich, parsimonious, and tractable.

Generative Modeling

- Treat the data points as **samples** from an unknown distribution



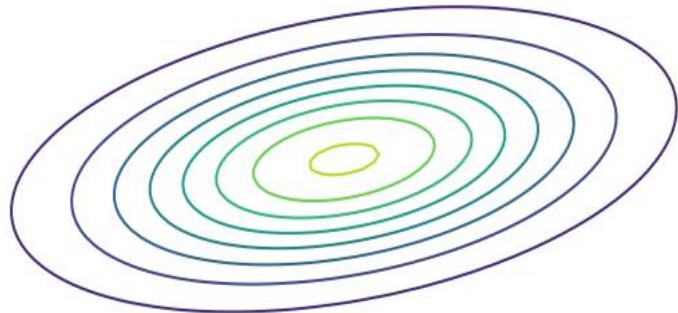
$$p^*(\mathbf{x})$$

$$\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\} \sim p^*(\mathbf{x})$$

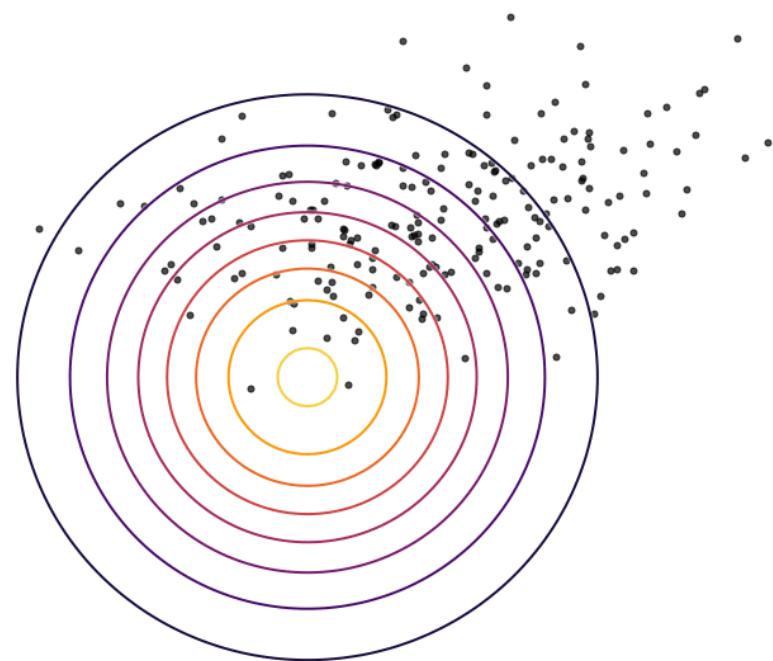
<https://colinraffel.com/blog/gans-and-divergence-minimization.html>

Generative Modeling

- Treat the data points as **samples** from an unknown distribution
- Design a **parameterized** distribution
- And **learn** its parameters to fit the samples



$p^*(x)$

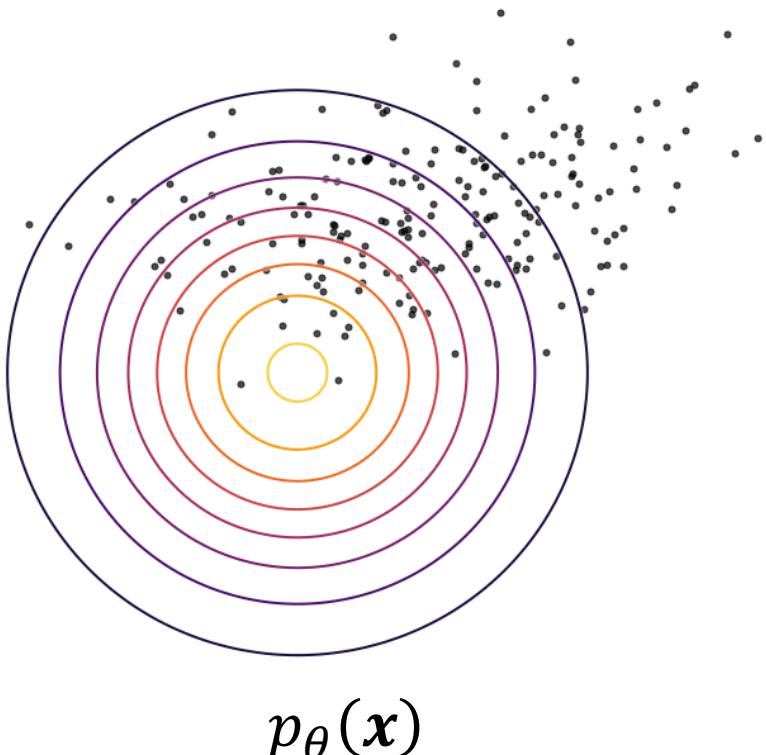


$p_\theta(x)$

<https://colinraffel.com/blog/gans-and-divergence-minimization.html>

Generative Modeling

$$\boldsymbol{x}^{(i)} \in \mathbb{R}^2$$

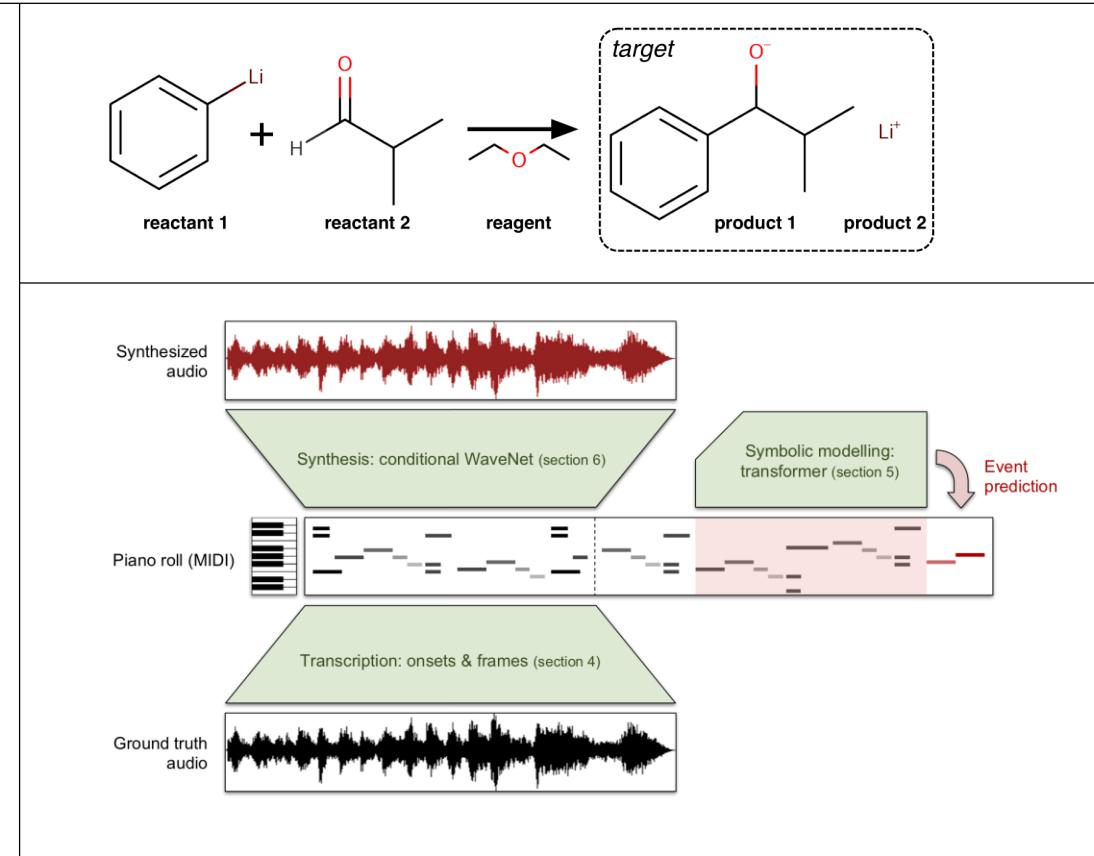
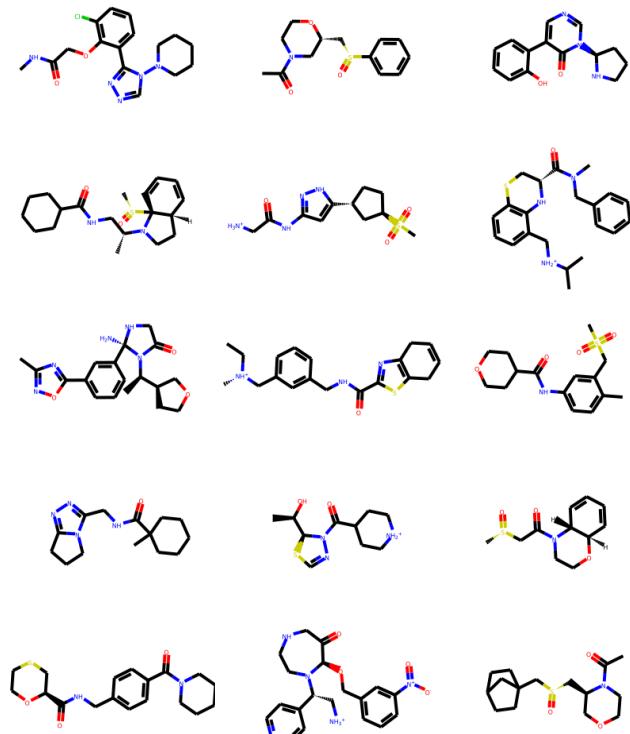


But $\boldsymbol{x}^{(i)}$ can also be:

- An **image** $\in \{0, \dots, 255\}^{W \times H \times 3}$
- A **sequence**
 - sentence, music notes, code
 - time series, audio, video
- A **graph**
 - molecular graph, AST, NN (!!!)
- A **set** (e.g., point cloud)
- A **record** in DB
- A **distribution**
- A **function**

Applications of Generative Models

- Main uses of a distribution: (1) **sampling**, (2) **density estimation**
 - **Generate** novel samples & **Evaluate** the probability density of a given sample



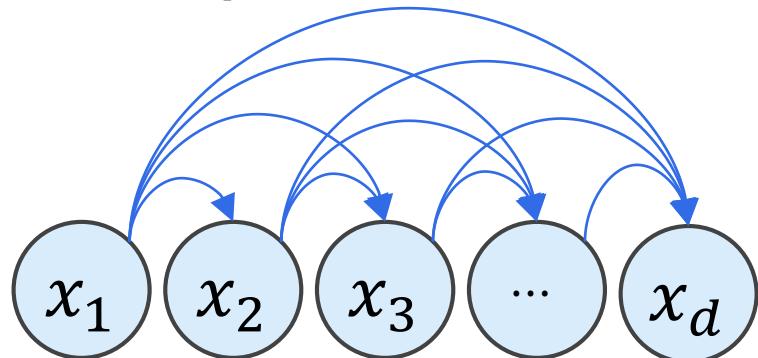
A Generative Model For Electron Paths. John Bradshaw, et al. ICLR 2019
Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. Curtis Hawthorne, et al. ICLR 2019

Outline

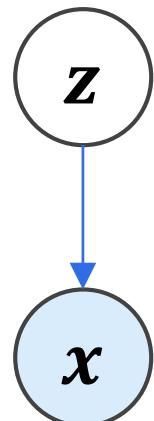
- What, Why
- **Landscape of DGMs**
- Sequential DGMs
- Summary & References

Landscape of (Deep) Generative Models

Autoregressive Models



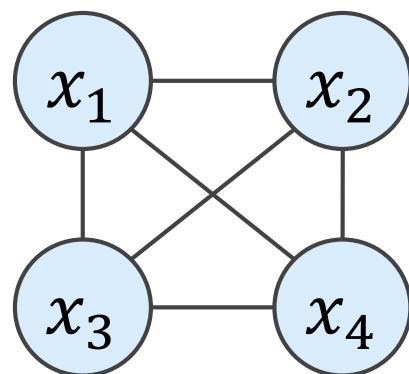
$$p_{\theta}(x) = \prod_{i=1}^d p_{\theta}(x_i | x_{<i})$$



$$p_{\theta}(x) = \int p_{\theta}(x|z)p_{\theta}(z) dz$$

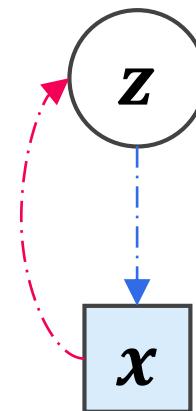
Latent Variable Models

Energy-Based Models



$$p_{\theta}(x) = \frac{e^{-E_{\theta}(x)}}{\int e^{-E_{\theta}(x)} dx}$$

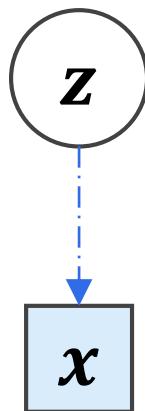
Normalizing Flows



$$x = f_{\theta}(z)$$

$$p_{\theta}(x) = p_{\theta}(z) \left| \det \frac{\partial f_{\theta}^{-1}(x)}{\partial z} \right|$$

Implicit Models



$$p_{\theta}(x) = ?$$

Autoregressive Models

- Any distribution can be factorized into an AR form
- Exact likelihood, easy to train
- Very generic and powerful
- Modern NNs make the training parallelizable
 - WaveNet, Causal CNN, Transformer

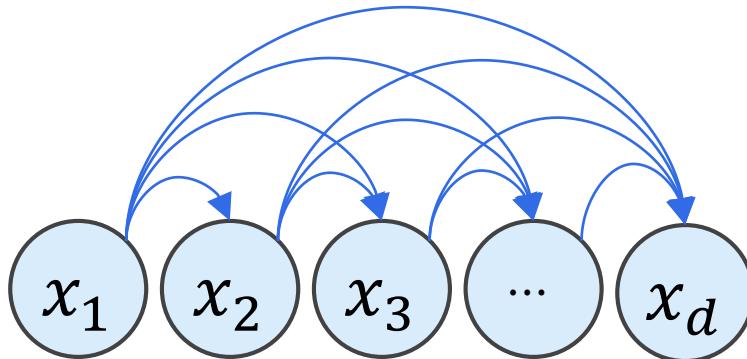
[**chain rule**]



Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization—all without task-specific training.

<https://openai.com/blog/better-language-models/>



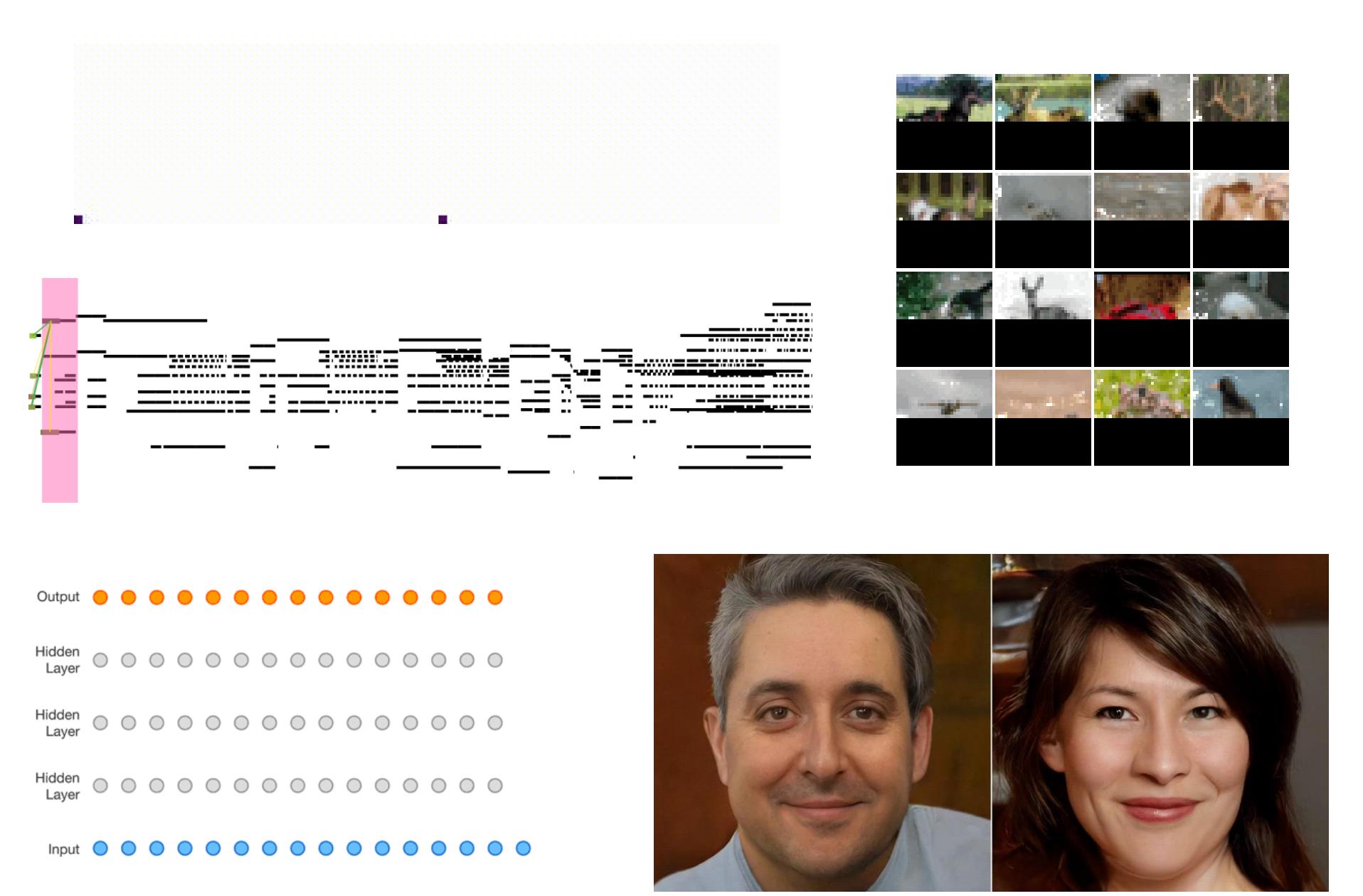
$$p_{\theta}(x) = \prod_{i=1}^d p_{\theta}(x_i | x_{<i})$$

$$p_{\theta}(x|c) = \prod_{i=1}^d p_{\theta}(x_i | x_{<i}, c)$$

(conditional, e.g., seq2seq)

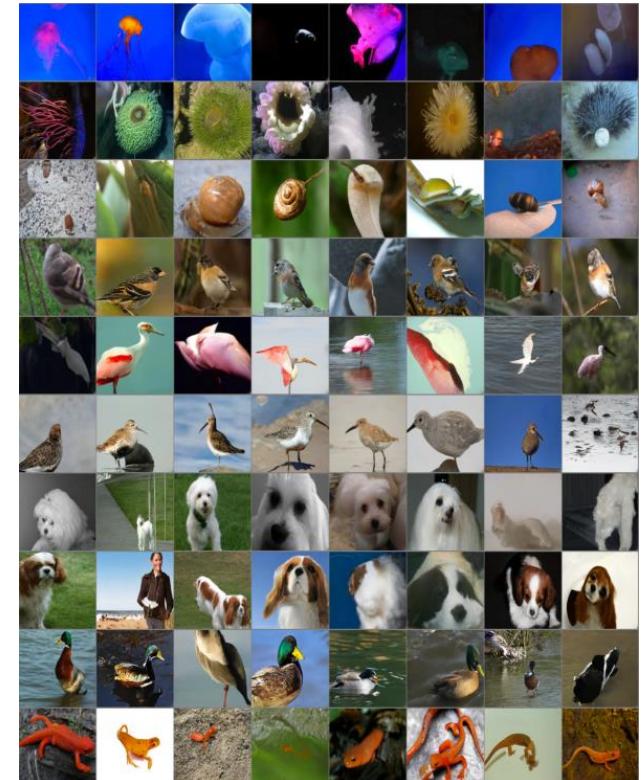
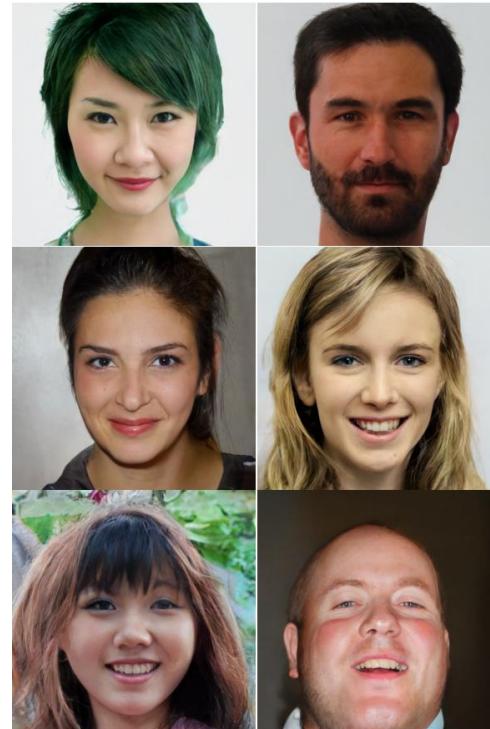
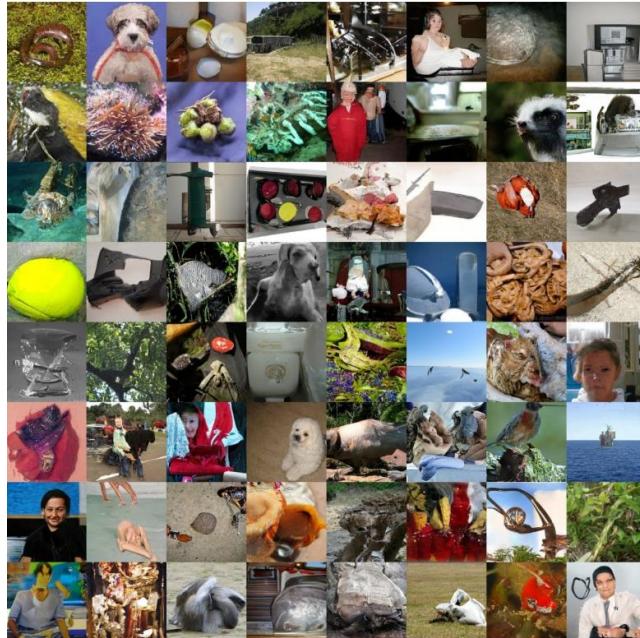


Hierarchical Autoregressive Image Models with Auxiliary Decoders.
Jeffrey De Fauw, et al. Arxiv 2019



Autoregressive Models

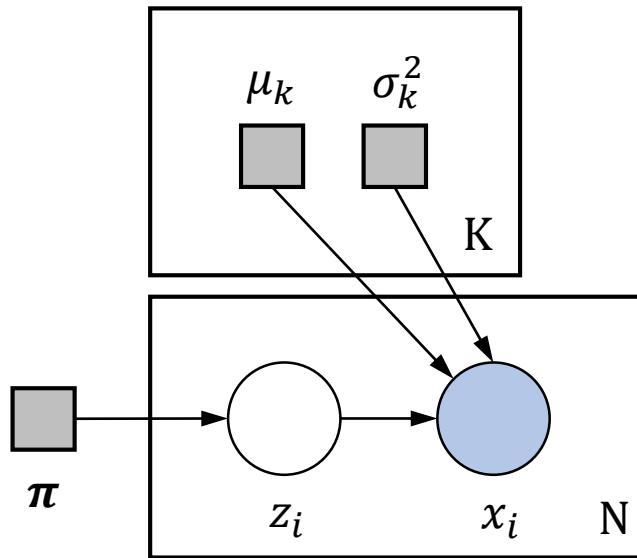
- More state-of-the-art (SOTA) results...
 - Also see Transformer-based AR models for text/audio/music/video/graph



Generating High Fidelity Images with Subscale Pixel Networks and Multidimensional Upscaling. Jacob Menick, Nal Kalchbrenner. ICLR 2019
Generating Diverse High-Fidelity Images with VQ-VAE-2. Ali Razavi, et al. Arxiv 2019

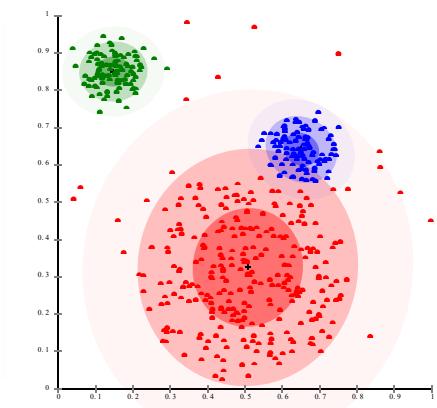
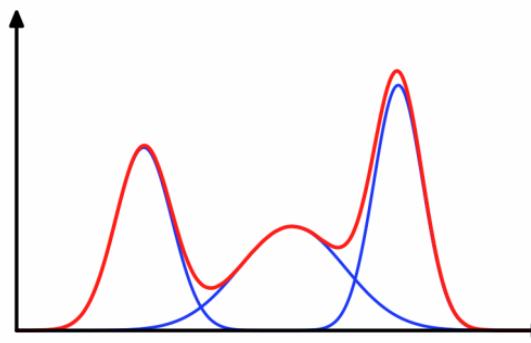
Latent Variable Models

- Example: Gaussian Mixture Models



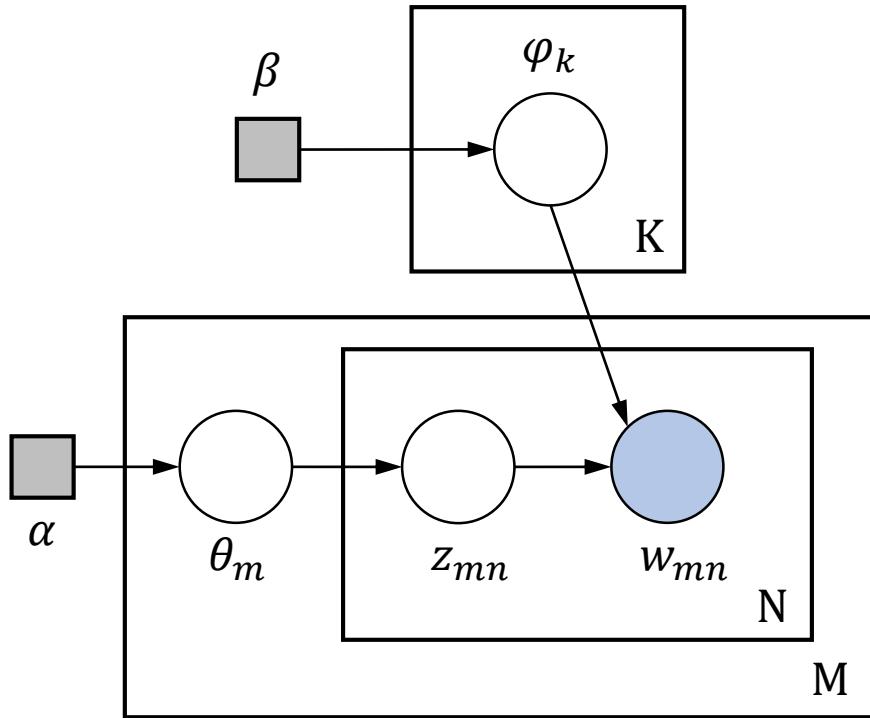
$$z_i \sim \text{categorical}(\pi_1, \dots, \pi_K), \quad i = 1, \dots, N,$$

$$x_i | z_i \sim \mathcal{N}(\mu_{z_i}, \sigma_{z_i}^2) \quad i = 1, \dots, N.$$



Latent Variable Models

- Example: Latent Dirichlet Allocation

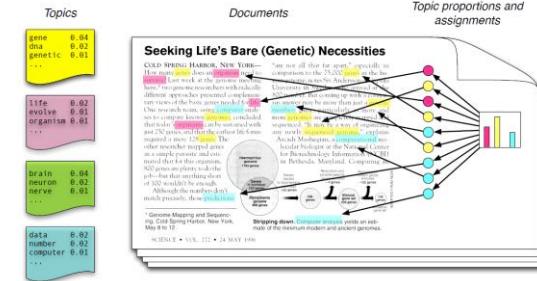


$$\varphi_k \sim Dir(\beta), \quad k = 1, \dots, K,$$

$$\theta_m \sim Dir(\alpha), \quad m = 1, \dots, M,$$

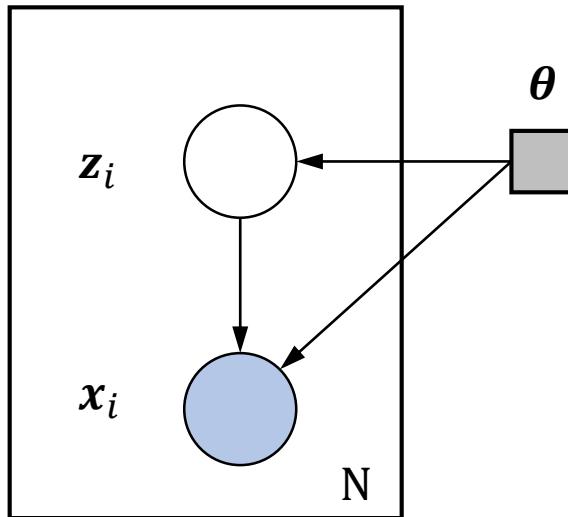
$$z_{mn} | \theta_m \sim \text{categorical}(\theta_m),$$

$$w_{mn} | \varphi, z_{mn} \sim \text{categorical}(\varphi_{z_{mn}})$$



Latent Variable Models

- Example: Deep Latent Gaussian Models



$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbb{I}), \quad i = 1, \dots, N,$$

$$x_i | \mathbf{z}_i \sim \mathcal{N}(\mu(\mathbf{z}_i), \sigma^2(\mathbf{z}_i)\mathbb{I}), \quad i = 1, \dots, N.$$

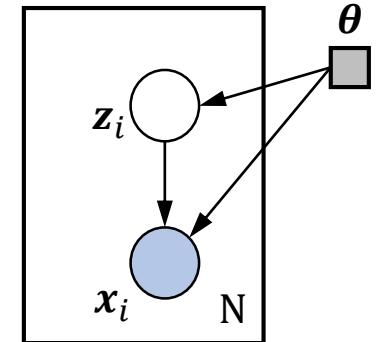
where $\mu(\cdot)$ and $\sigma(\cdot)$ are neural networks,
 $\theta = \{\text{parameters of } \mu \text{ and } \sigma\}$



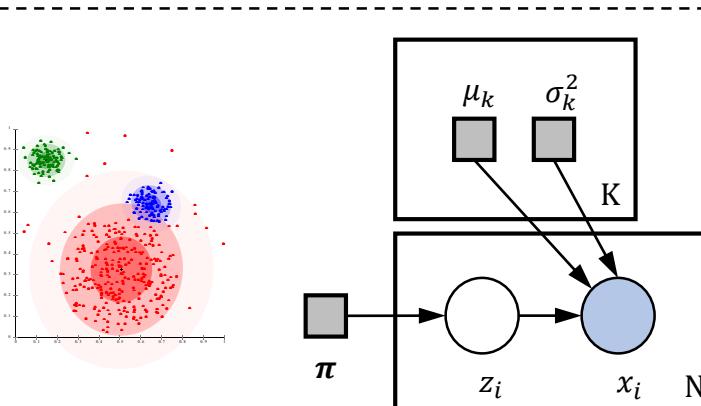
BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. Lars Maaløe, et al. Arxiv 2019

Latent Variable Models

- Learning: fit the model to data
 - Find the maximum likelihood estimation of the parameters θ
 - Inference: compute unknown probability distributions
 - Posterior distribution of latent variable z
 - Marginal likelihood of observations x



$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})}$$

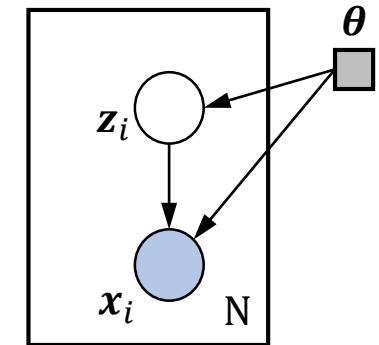


LVM: The Need for **Approximate** Inference

- **Posterior** distribution of latent variables $\mathbf{z} = \{\mathbf{z}_i\}, i = 1 \dots M$
- **Marginal** likelihood of observations $\mathbf{x} = \{x_i\}, i = 1 \dots N$

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}}$$

likelihood prior
marginal/evidence



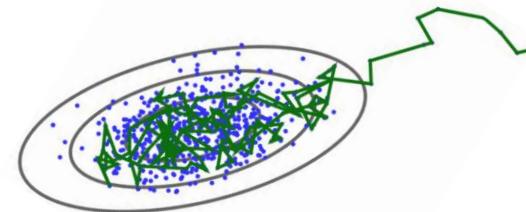
For many models, the integral (or sum) is **intractable**:

- Unavailable in closed form, or
- Requires exponential time to compute.

LVM: Approximate Inference of $p(\mathbf{z}|\mathbf{x})$

- Sampling: MCMC (Metropolis-Hastings or Gibbs sampling)

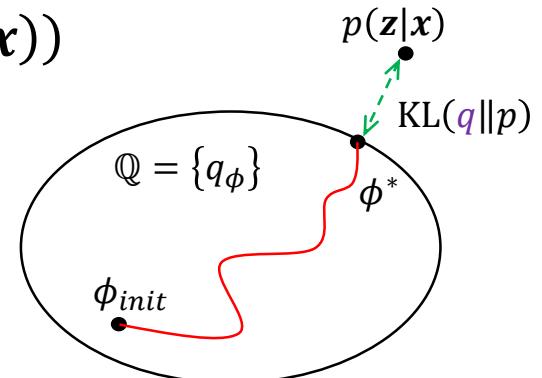
- Approximate the posterior using samples
- ✓ Converge to the posterior asymptotically
- ✗ Computationally intensive



- Variational Inference: turn inference into an optimization problem
 - Set up a **family** of approximate densities \mathbb{Q} over the latent variables
 - Find the member q^* in the family \mathbb{Q} that is closest to the exact posterior

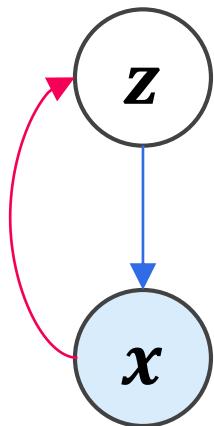
$$q^*(\mathbf{z}) = \arg \min_{q(\mathbf{z}) \in \mathbb{Q}} \text{KL}(q(\mathbf{z}) \| p(\mathbf{z}|\mathbf{x}))$$

- ✓ Faster and easier to scale to large datasets
- ✗ Biased

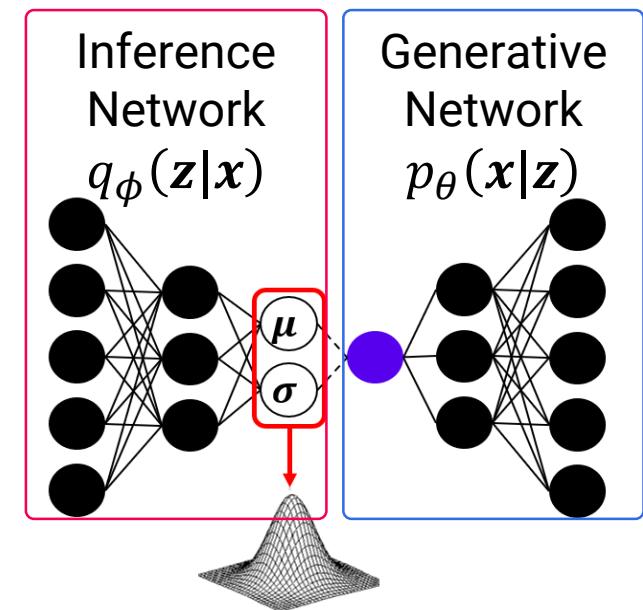


LVM: Variational Autoencoders

- VAE uses *amortized VI* by introducing a parameterized ***inference network*** (i.e., encoder)
- Parameters of the encoder and the decoder are jointly optimized
 - By SGD on the Evidence Lower Bound Objective (ELBO)
- Make the probabilistic graphical models great again
 - Enable us to build PGMs with neural nets in a flexible way

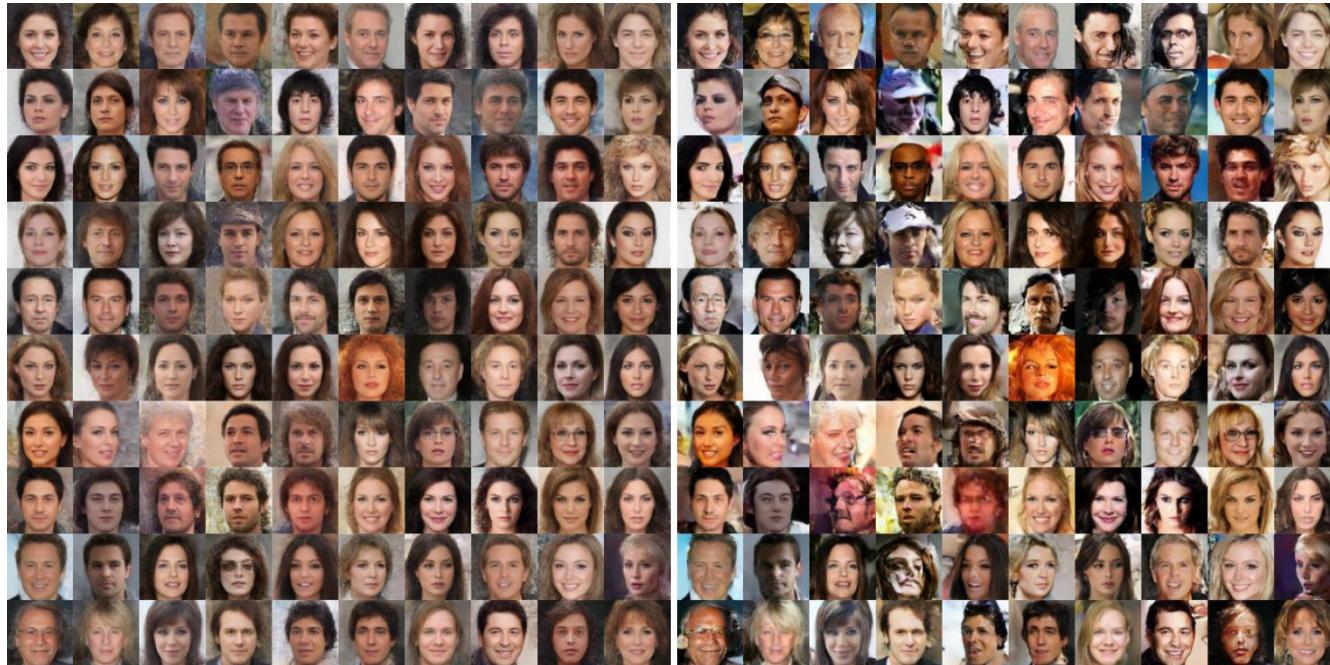


$$p_{\theta}(x) = \int p_{\theta}(x, z) dz \\ \geq \mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(z, x)}{q_{\phi}(z|x)} \right]$$



Latent Variable Models

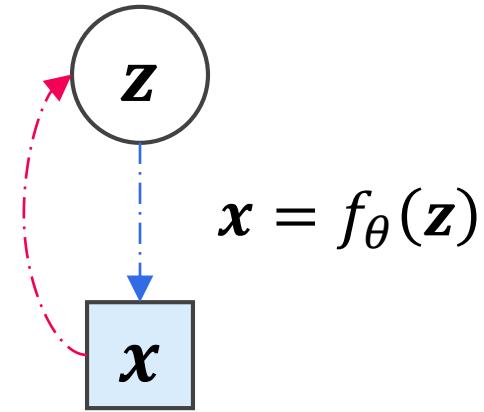
- SOTA VAEs are lagging behind AR models on image generation
- But deep SSMs (e.g., stochastic RNNs) tend to outperform RNN-based AR models on datasets that exhibit natural sequential order
 - More on this later



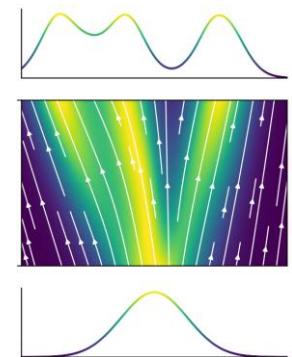
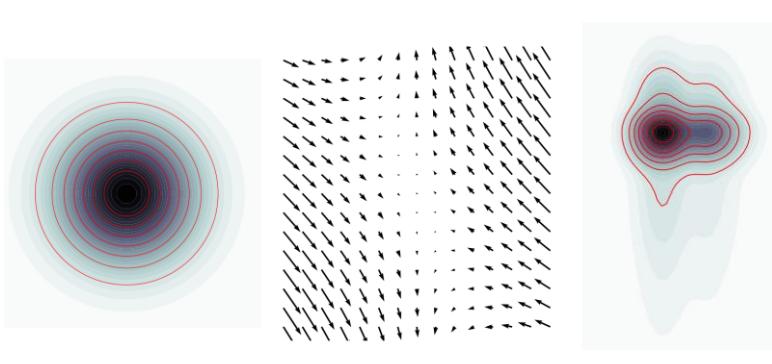
BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. Lars Maaløe, et al. Arxiv 2019

Normalizing Flows

- Transform a simple density to a complex one
 - Using multilayer **invertible** mappings (or ODEs)
- Exact likelihood by the **change of variable** formula
- Usually fast sampling
- Exciting new area, many interesting ideas
 - Bonus: can help reduce memory footprint



$$p_\theta(x) = p_\theta(z) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$



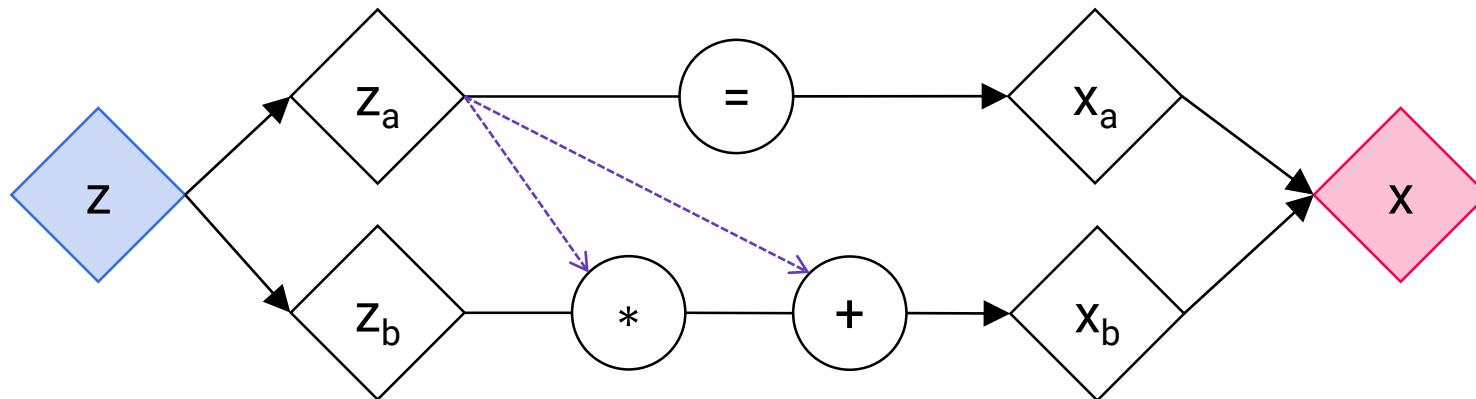
A Tutorial on Deep Probabilistic Generative Models. Ryan P. Adams. Machine Learning Summer School 2018
FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. W Grathwohl, et al. ICLR 2019

Normalizing Flows

- Example: Design an invertible mapping $f_\theta: \mathbb{R}^D \rightarrow \mathbb{R}^D$ [coupling layer]

$$z_{1:d}^{(l+1)} = z_{1:d}^{(l)}$$

$$z_{d+1:D}^{(l+1)} = z_{d+1:D}^{(l)} \odot \exp\left(s_\theta\left(z_{1:d}^{(l)}\right)\right) + t_\theta\left(z_{1:d}^{(l)}\right)$$

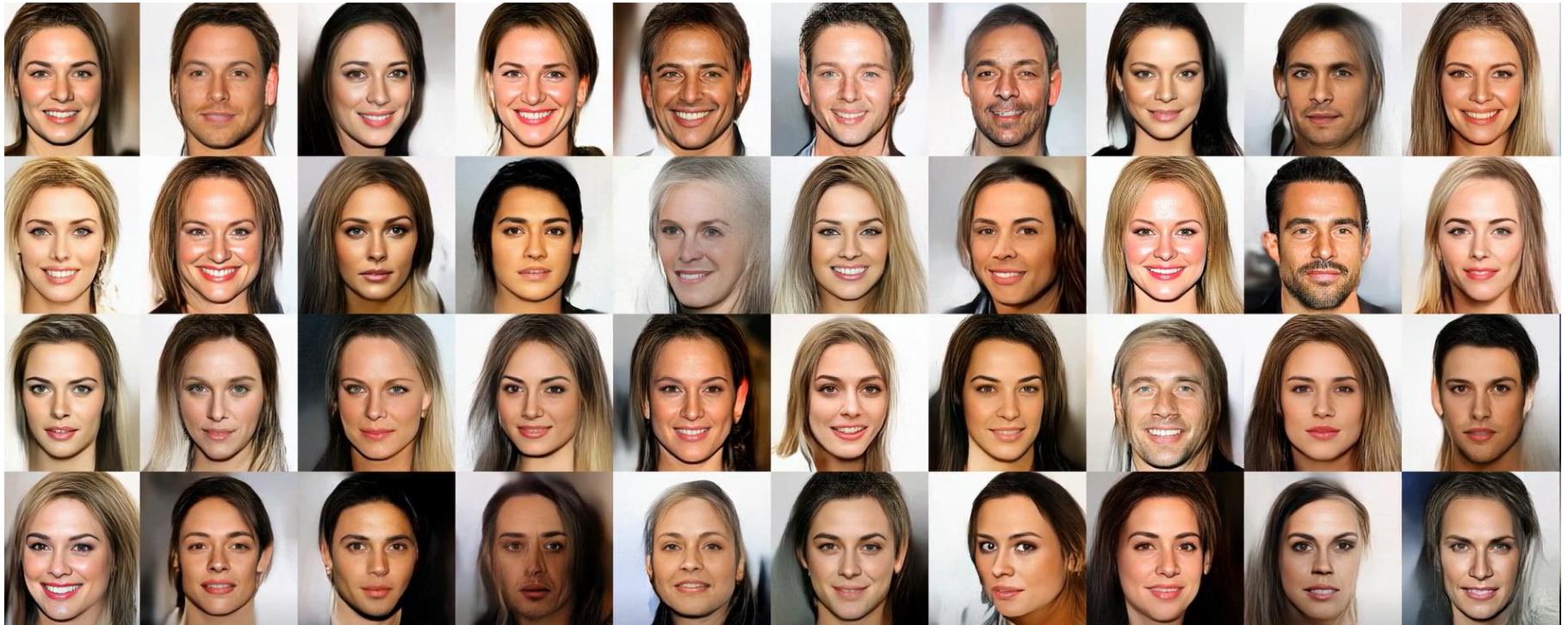


$$\log \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right| = - \sum_{i=1}^{D-d} s_\theta(z_a)_i$$

Density Estimation using Real NVP. L Dinh, et al. ICLR 2017

Normalizing Flows

- Recent advances make them comparable to AR models
 - Lagging behind AR on unconditional generation when trained on ImageNet

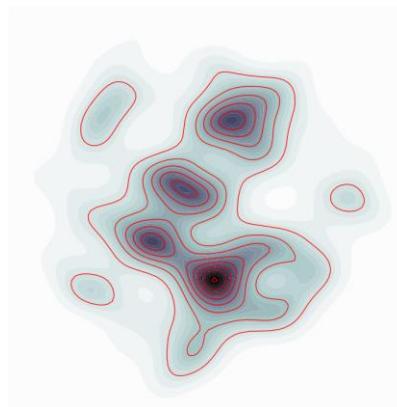
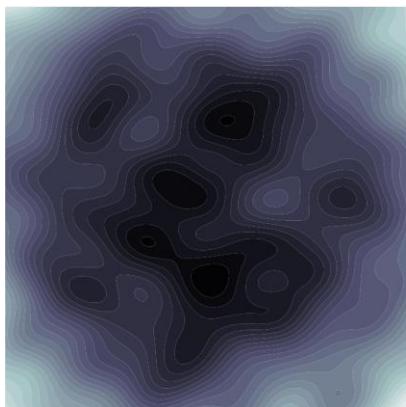


Glow: Generative Flow with Invertible 1x1 Convolutions. DP Kingma and P Dhariwal. NeurIPS 2018

Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. Jonathan Ho, et al. ICML 2019

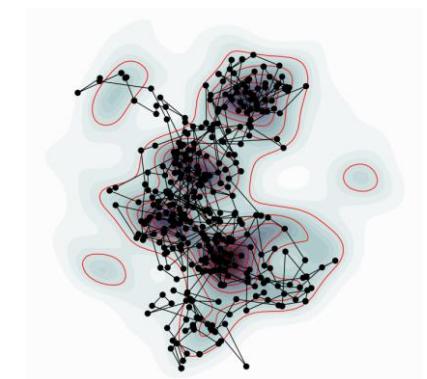
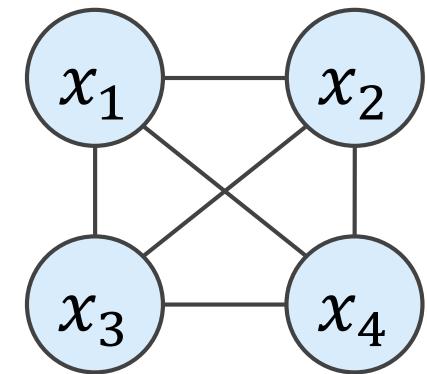
Energy-Based Models

- Parametrize any scalar **energy** function $E_\theta(x)$
 - $-E_\theta(x)$ can be interpreted as a **score** function
 - i.e., it gives the “goodness of configurations”
- Intractable likelihood, hard to train
 - Because the partition function Z_θ is intractable
- Sampling is usually hard and slow (need MCMC)
- Highly expressive and flexible
 - Also some nice properties, e.g., compositionality



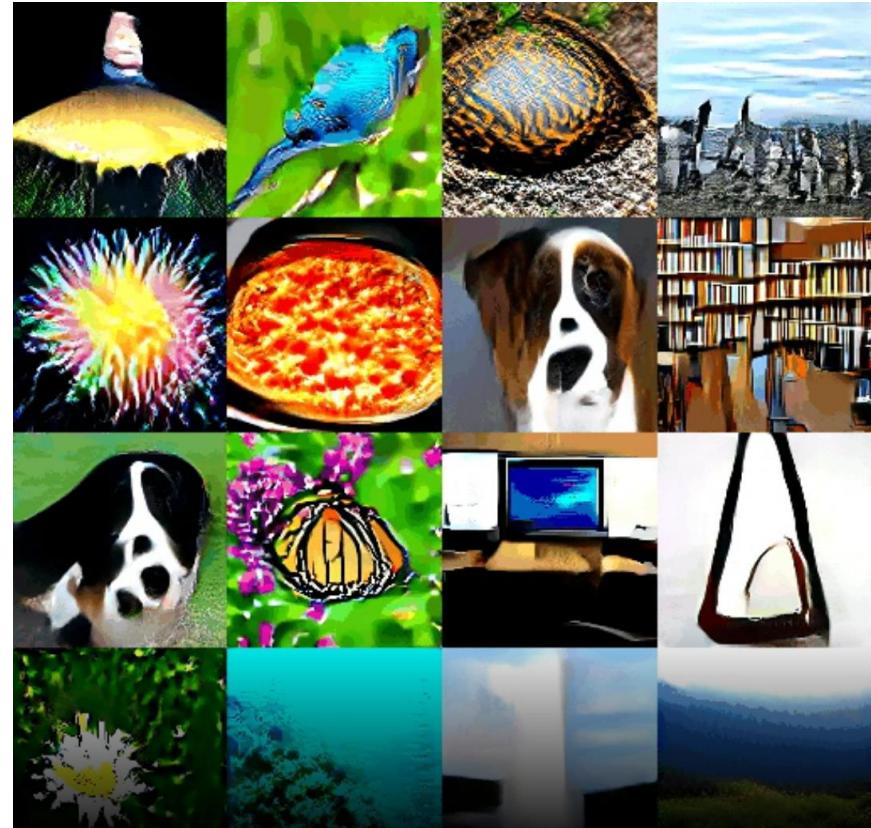
$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}$$

$$Z_\theta = \int e^{-E_\theta(x)} dx$$



Energy-Based Models

- There is renewed interest in them; but still have a long way to go



<https://openai.com/blog/energy-based-models/>

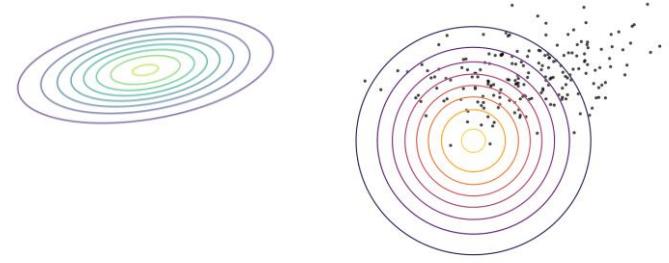
Maximum Entropy Generators for Energy-Based Models. Rithesh Kumar, et al. Arxiv 2019

On the Anatomy of MCMC-based Maximum Likelihood Learning of Energy-Based Models. Erik Nijkamp, et al. CVPR 2019

Quick Summary

- Where are (your favorite) **GANs**? Wait a minute ...
- In generative modeling, our goal is ...
to make the model $p_\theta(x)$ **mimic** the true data distribution $p^*(x)$
 - By minimizing some **divergence** measure: $\theta^* = \operatorname{argmin}_\theta D(p^*, p_\theta)$
- So far, AR/Flow/LVM/EBM are usually trained via MLE
 - Why MLE? It was derived by choosing D to be **Kullback–Leibler divergence**

$$\begin{aligned}\theta^* &= \operatorname{argmin}_\theta \mathbf{KL}(p^* \| p_\theta) = \operatorname{argmin}_\theta \mathbb{E}_{x \sim p^*} \left[\log \frac{p^*(x)}{p_\theta(x)} \right] \\ &= \operatorname{argmin}_\theta \mathbb{E}_{x \sim p^*} [\log p^*(x)] - \mathbb{E}_{x \sim p^*} [\log p_\theta(x)] \\ &= \operatorname{argmax}_\theta \mathbb{E}_{x \sim p^*} [\log p_\theta(x)] \approx \operatorname{argmax}_\theta \sum_{i=1}^N \log p_\theta(x^{(i)})\end{aligned}$$



<https://colinraffel.com/blog/gans-and-divergence-minimization.html>

Quick Summary

- Some generated images are not so realistic... Why?

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathbf{KL}(p^* \| p_{\theta}) \approx \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^N \log p_{\theta}(x^{(i)})$$

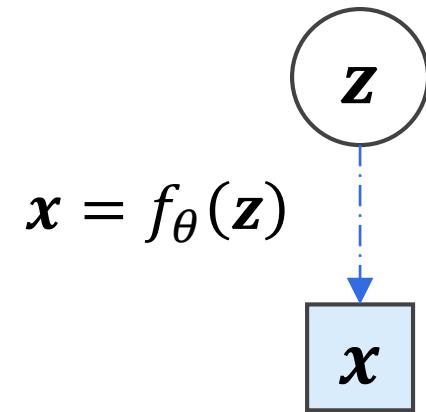
- For real data, most models are mis-specified: $p^* \notin \{p_{\theta} | \theta \in \Theta\}$



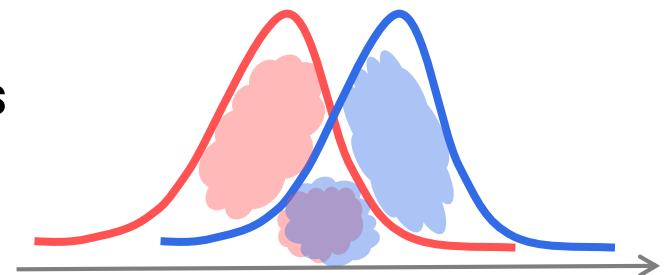
<https://colinraffel.com/blog/gans-and-divergence-minimization.html>

Implicit Models

- A.k.a. ***likelihood-free*** models or ***simulator*** models
 - Also called ***neural samplers*** when $f_\theta(\cdot)$ is NN
- No explicit likelihood, so MLE doesn't work
 - Because $f_\theta(\cdot)$ may be non-surjective and non-injective
- Can only sample from the model
 - Sampling is fast and differentiable w.r.t. θ
- I just want fantasy samples. How to learn the model?
- Idea: comparing p_θ and p^* using ***samples***

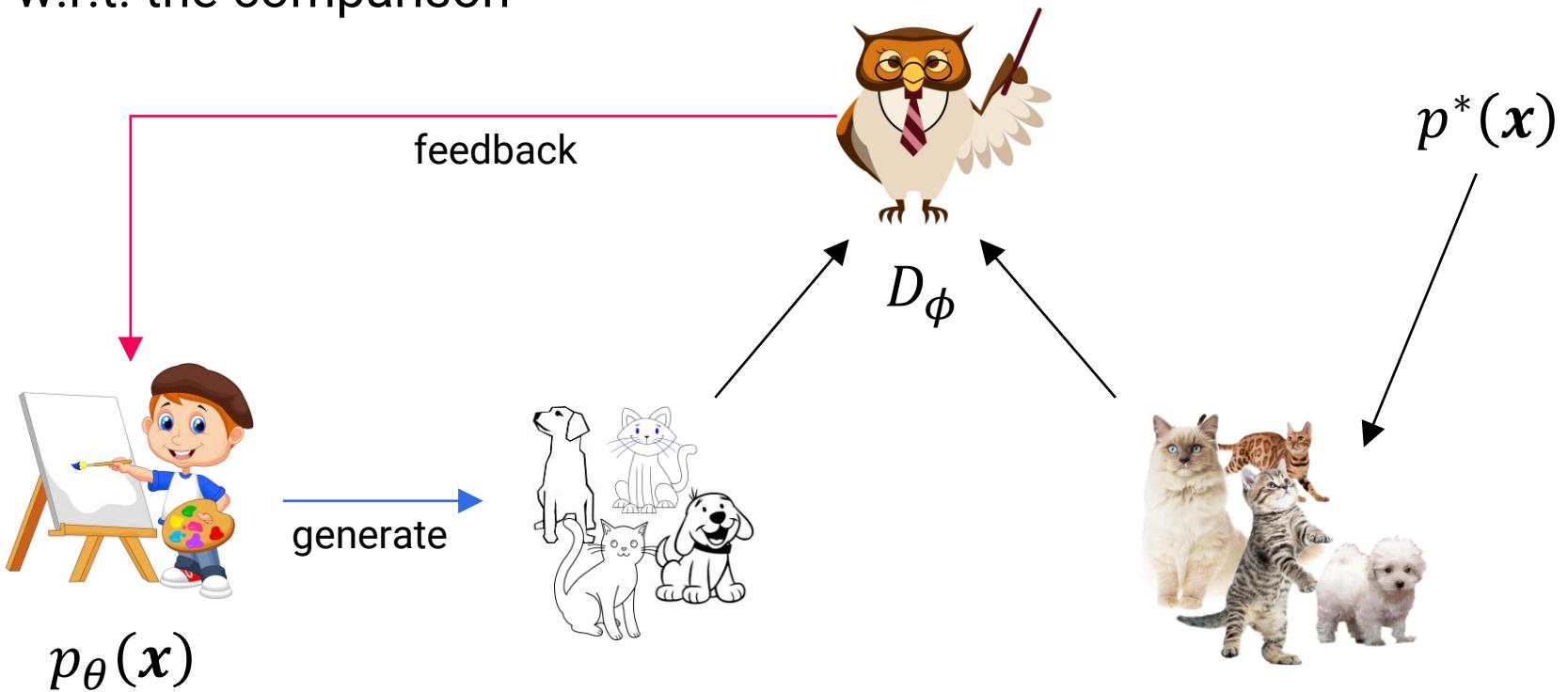


$$p_\theta(x) = ?$$



Implicit Models: Learning by Comparison

- Use an auxiliary model f_ϕ (a.k.a. discriminator/critic/teacher) to **test** how a set of simulated samples differs from observed data
- Adjust the generative model p_θ to better match the data distribution w.r.t. the comparison



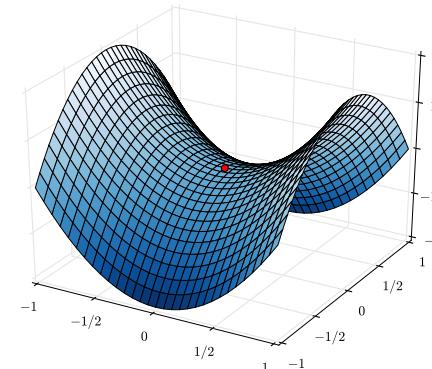
The Maximum Mean Discrepancy for Training Generative Adversarial Networks. Arthur Gretton. 2018

Implicit Models: GAN

- The teacher should be strict
- The student should challenge the teacher
- Let them play the following two-player minimax game:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p^*} [\log D_{\phi}(x)] + \mathbb{E}_{x \sim p_{\theta}} [\log (1 - D_{\phi}(x))]$$

- Implement this game using ***alternating optimization***
 - Alternate between (1 step of) updating θ and (k step of) updating ϕ
 - One player minimizes the objective function maximized by the other
 - Seek a **saddle point** / an **equilibrium**



GAN: Mode Collapse

- GAN is shown to be minimizing the **Jensen–Shannon divergence** or **reverse Kullback–Leibler divergence** between p^* and p_θ

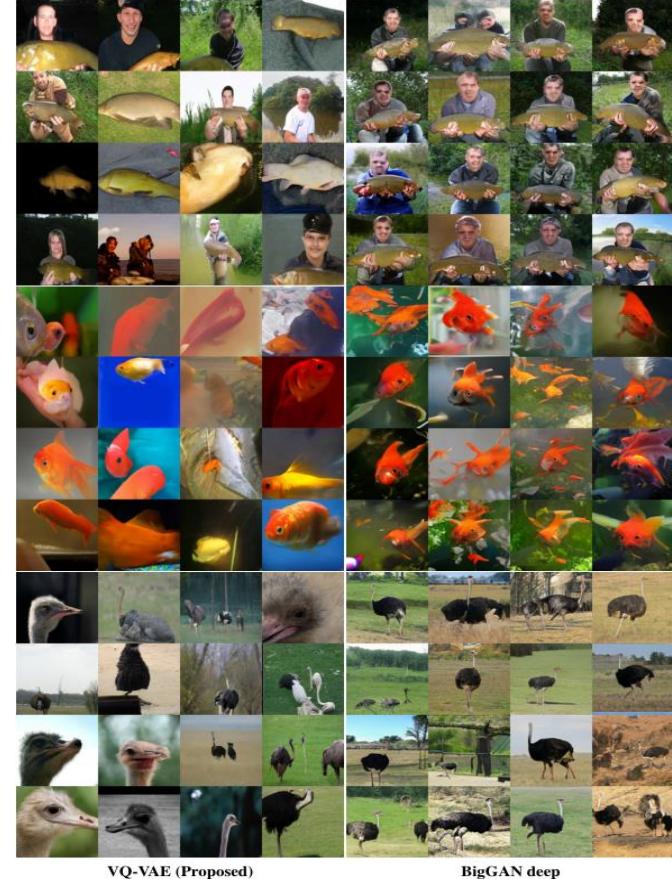
$$\min_{\theta} \text{KL}(p_\theta \| p^*) = \min_{\theta} \mathbb{E}_{x \sim p_\theta} \left[\log \frac{p_\theta(x)}{p^*(x)} \right]$$



<https://colinraffel.com/blog/gans-and-divergence-minimization.html>

Implicit Models

- SOTA GANs can generate fantasy samples
 - Trade-off between *quality* & *diversity*



Training Set	Resolution	Top-5 Accuracy	Top-1 Accuracy	Inception Score	FID-50K
Real BigGAN-deep Hierarchical Autoregressive	128×128	88.79%	68.82%	165.38 ± 2.84	1.61
	128×128	64.44%	40.64%	71.31 ± 1.57	4.22
	128×128	77.33%	54.05%	17.02 ± 0.79	46.05
Real BigGAN-deep High-Res VQ-VAE	256×256	91.47%	73.09%	331.83 ± 5.00	2.47
	256×256	65.92%	42.65%	109.39 ± 1.56	11.78
	256×256	77.59%	54.83%	43.44 ± 0.87	38.05

Large Scale GAN Training for High Fidelity Natural Image Synthesis. Andrew Brock, et al. ICLR 2019
Classification Accuracy Score for Conditional Generative Models. Suman Ravuri, Oriol Vinyals. Arxiv 2019
Generating Diverse High-Fidelity Images with VQ-VAE-2. Ali Razavi, et al. Arxiv 2019

Quick Summary

	Sampling	Likelihood	Training	Latent	SOTA Perf.
AR	Slow	Exact	MLE	No*	1*
Flow	Fast*	Exact	MLE	Yes	2
LVM	Fast	Approx.	Approx. MLE	Yes	3
EBM	Very slow	Intractable*	Approx. MLE	No	4
Implicit	Fast	Intractable	Adversarial	No*	1*

Outline

- What, Why
- Landscape of DGMs
- **Sequential DGMs**
- Summary & References

DGMs for Sequential Data

- For sequential data, we may:
 - Simply use AR models, as they perfectly match the order of data
 - Simply plug RNNs/CNNs into VAEs/GANs and play
- Besides, we can further exploit the sequential structure to design more flexible/expressive sequential (D)GMs



A Spectrum of Time Series Models

- Interestingly, most time series models are generative models

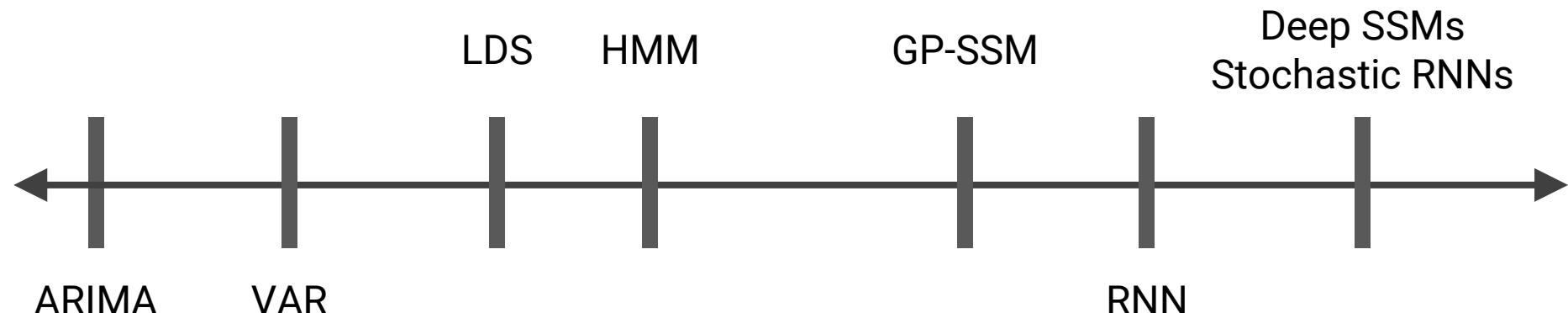
Limited capacity

Data efficient

Fast learning/inference

Easy to understand

Highly flexible
(Data intensive)
Expensive learning
(Harder to interpret)



ARIMA

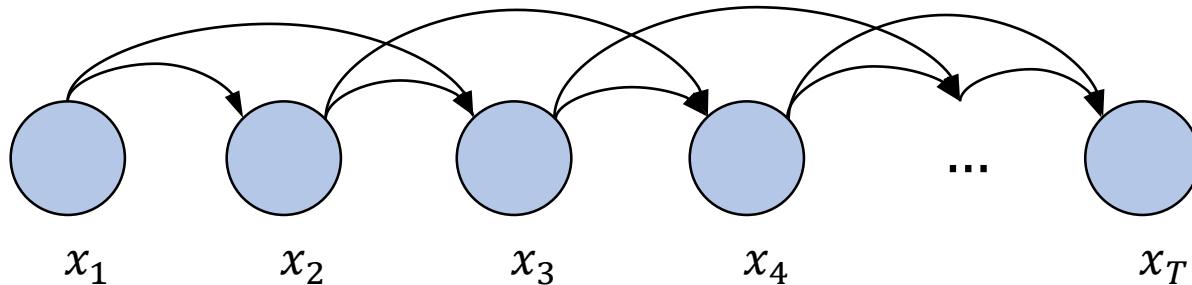
- Example 1: Auto-Regressive Integrated Moving Average

$$\text{ARIMA}(p, 0, 0): \quad x_t = c + \epsilon_t + \sum_{i=1}^p \theta_i x_{t-i}, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

\Leftrightarrow

$$x_t \sim \mathcal{N}\left(c + \sum_{i=1}^p \theta_i x_{t-i}, \sigma^2\right)$$

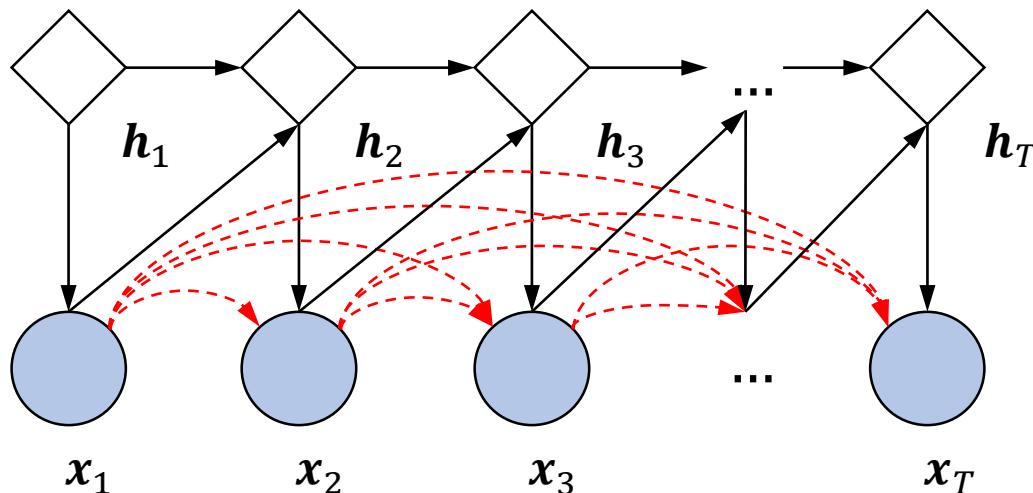
$$p_\theta(x_{1:T}) = p_\theta(x_{1:p}) \prod_{t=p+1}^T p_\theta(x_t | x_{t-p:t-1})$$



RNN

- Example 2: Recurrent Neural Networks (LSTM, GRU, ...)
- RNNs are also **autoregressive** models
 - Compared to ARIMA, RNNs enable longer and more complex historical dependencies

$$p_{\theta}(x_{1:T}) = \prod_{t=1}^T p_{\theta}(x_t | \mathbf{x}_{<t}) = \prod_{t=1}^T p_{\theta}(x_t | \mathbf{h}_t)$$



$$x_1 \sim g_{\theta}(\cdot | \mathbf{h}_1)$$

$$\mathbf{h}_t = \text{LSTM}_{\theta}(\mathbf{h}_{t-1}, x_{t-1})$$

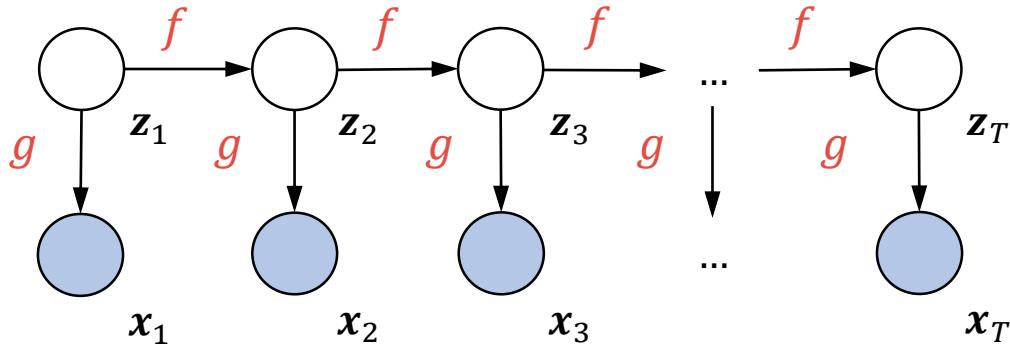
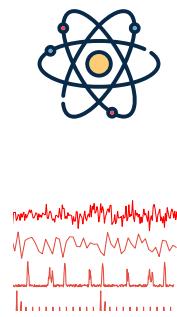
$$x_t | x_{1:t-1} \sim g_{\theta}(\cdot | \mathbf{h}_t)$$

where $t = 2 \dots T$

the actual conditional dependences
are shown in red dashed lines

State-Space Models

- SSMs are a general class of probabilistic sequential models
 - Unlike AR models, SSMs introduce latent variables to enhance their capability
- Basic elements in SSMs:
 - parameters θ , prior density π_θ , transition density f_θ , measurement density g_θ
 - latent states \mathbf{z}_t , observations \mathbf{x}_t

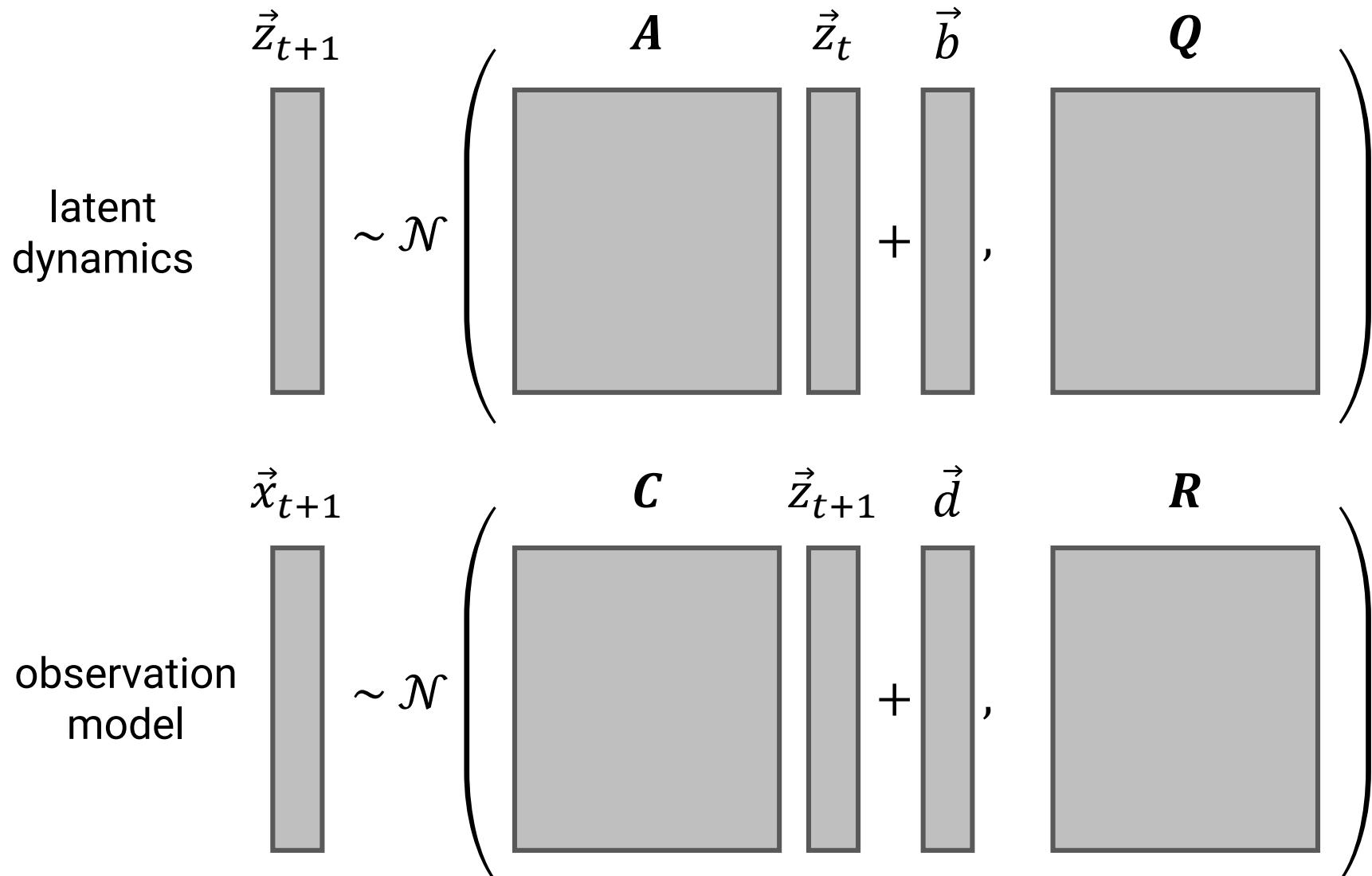


$$\begin{aligned}\mathbf{z}_1 &\sim \pi_\theta(\cdot) \\ \mathbf{z}_t | \mathbf{z}_{t-1} &\sim f_\theta(\cdot | \mathbf{z}_{t-1}) \\ \mathbf{x}_t | \mathbf{z}_t &\sim g_\theta(\cdot | \mathbf{z}_t)\end{aligned}$$

where $t = 2 \dots T$

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \pi_\theta(\mathbf{z}_1) g_\theta(\mathbf{x}_1 | \mathbf{z}_1) \prod_{t=2}^T f_\theta(\mathbf{z}_t | \mathbf{z}_{t-1}) g_\theta(\mathbf{x}_t | \mathbf{z}_t)$$

Example: Gaussian Linear Dynamical Systems



Example: Deep SSMs

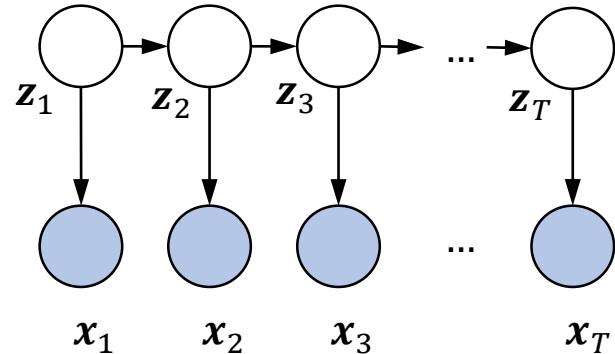
- Deep Markov Model

$$\mathbf{z}_1 \sim \pi_\theta(\cdot)$$

$$\mathbf{z}_t | \mathbf{z}_{t-1} \sim \mathcal{N}\left(\mu_\theta(\mathbf{z}_{t-1}), \sigma_\theta^2(\mathbf{z}_{t-1})\right)$$

$$x_t | \mathbf{z}_t \sim g_\theta(\cdot | \mathbf{z}_t)$$

where $\mu_\theta(\cdot)$ and $\sigma_\theta^2(\cdot)$ are (gated) neural networks



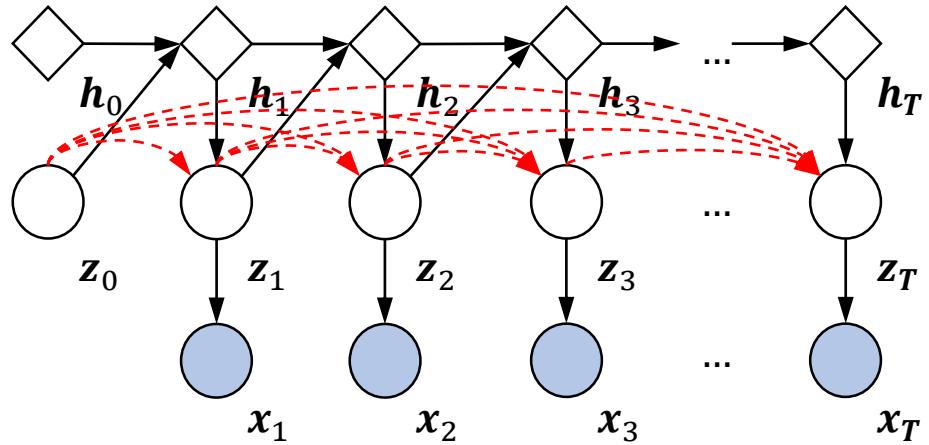
- LSTM State Space Model

$$\mathbf{z}_0 \sim \pi_\theta(\cdot)$$

$$\mathbf{h}_t = \text{LSTM}_\theta(\mathbf{h}_{t-1}, \mathbf{z}_{t-1})$$

$$\mathbf{z}_t | \mathbf{z}_{1:t-1} \sim f_\theta(\cdot | \mathbf{h}_t)$$

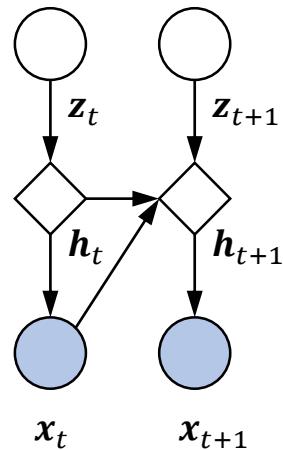
$$x_t | \mathbf{z}_t \sim g_\theta(\cdot | \mathbf{z}_t)$$



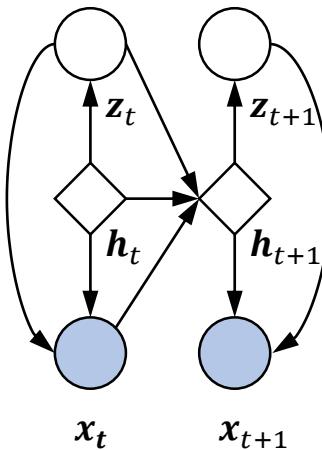
Example: Stochastic RNNs

- Injecting random variables into vanilla RNNs

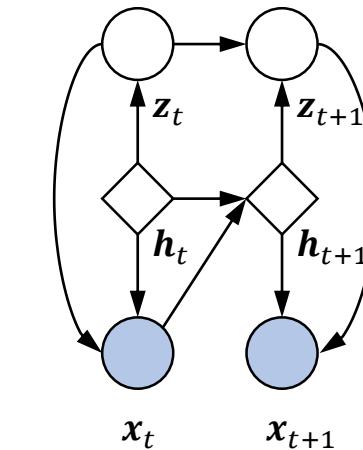
STORN [Arxiv 2014]
Noisin [ICML 2018]



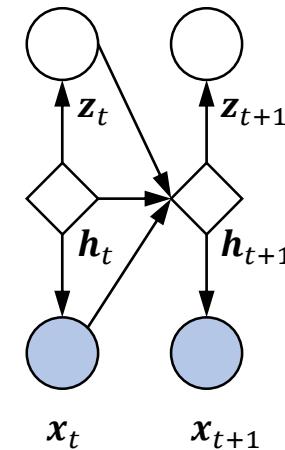
VRNN [NIPS 2015]



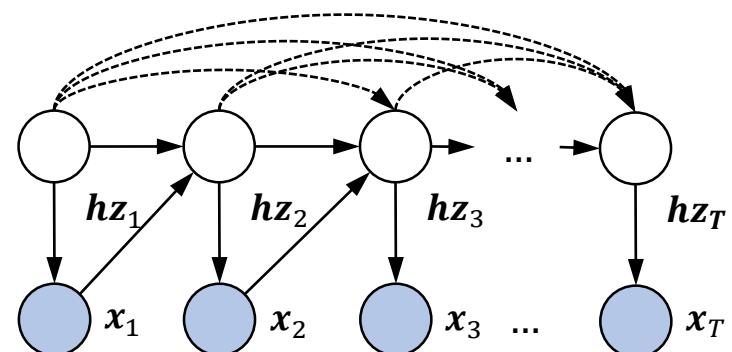
SRNN [NIPS 2016]



Z-Forcing [NIPS 2017]



After merging
the deterministic & stochastic state
at each time step,
almost all of them resemble this:

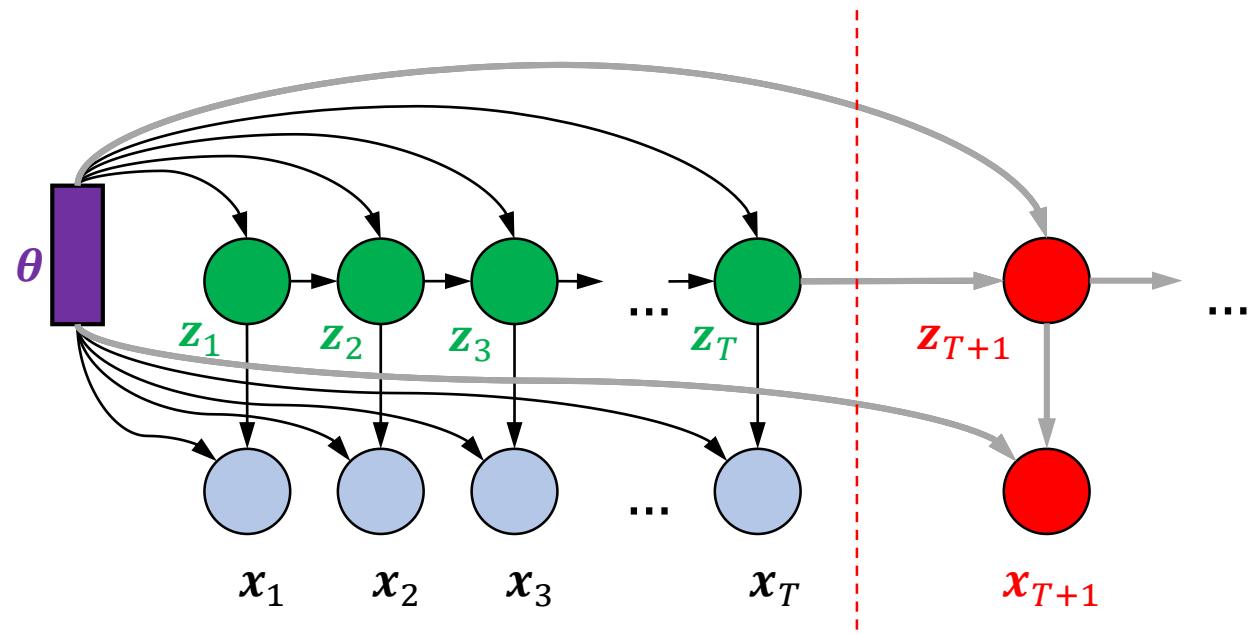


Playing with Probabilistic Sequential Models

- Given these models, our basic tasks are:
 - Model learning:** MLE for the model parameters θ : $\theta_{MLE} = \underset{\theta}{\operatorname{argmax}} p_{\theta}(x)$
 - Bayesian inference:** posterior distribution of latent variables z : $p_{\theta}(z_{1:T} | x_{1:T})$
 - Prediction:** conditional distribution of future states: $p_{\theta}(z_{T+1:T+L} | x_{1:T})$
 - And future observations: $p_{\theta}(x_{T+1:T+L} | x_{1:T})$

Many learning algorithms available:

- Expectation Maximization
- Variational Inference
- Sequential Monte Carlo
- Particle MCMC
- Likelihood-free methods



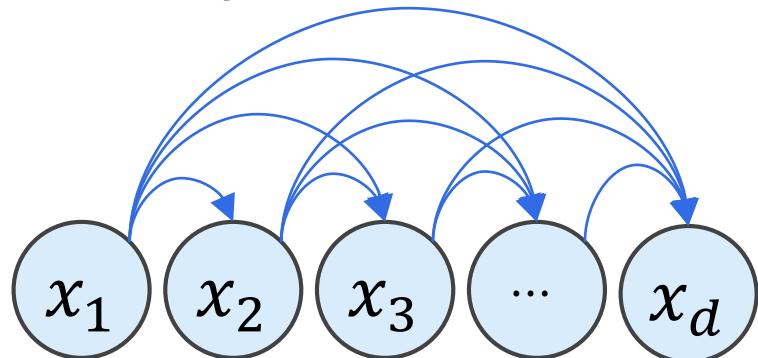
On Particle Methods for Parameter Estimation in State-Space Models. N Kantas, et al. Statistical Science 2015

Outline

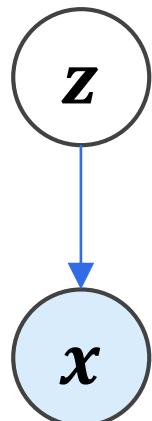
- What, Why
- Landscape of DGMs
- Sequential DGMs
- **Summary & References**

Landscape of (Deep) Generative Models

Autoregressive Models



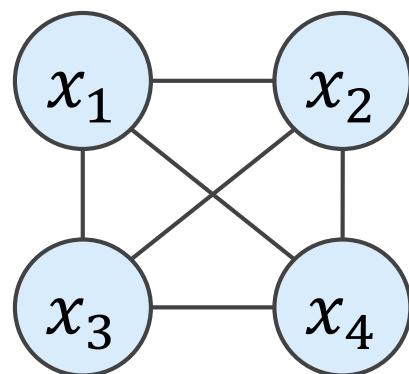
$$p_\theta(x) = \prod_{i=1}^d p_\theta(x_i | x_{<i})$$



$$p_\theta(x) = \int p_\theta(x|z)p_\theta(z) dz$$

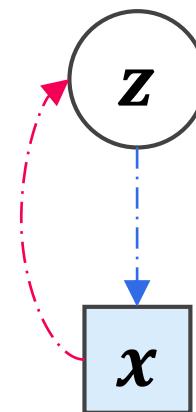
Latent Variable Models

Energy-Based Models



$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{\int e^{-E_\theta(x)} dx}$$

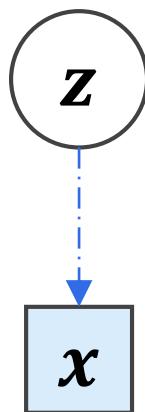
Normalizing Flows



$$x = f_\theta(z)$$

$$p_\theta(x) = p_\theta(z) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$

Implicit Models



$$p_\theta(x) = ?$$

Pros & Cons

	Sampling	Likelihood	Training	Latent	SOTA Perf.
AR	Slow	Exact	MLE	No*	1*
Flow	Fast*	Exact	MLE	Yes	2
LVM	Fast	Approx.	Approx. MLE	Yes	3
EBM	Very slow	Intractable*	Approx. MLE	No	4
Implicit	Fast	Intractable	Adversarial	No*	1*

Courses & Tutorials & Monographs

Courses

- Berkeley CS294-158: Deep Unsupervised Learning, Spring 2019. [[link](#)]
- Stanford CS 236: Deep Generative Models, Fall 2018-2019. [[link](#)]
- Edinburgh: Probabilistic Modelling and Reasoning, 2018/19. [[link](#)]

Tutorials

- Tutorial on Deep Generative Models. Shakir Mohamed and Danilo Rezende. UAI 2017 & CCN 2018. [[slides 1](#)][[slides 2](#)][[video](#)]
- A Tutorial on Deep Probabilistic Generative Models. Ryan P. Adams. MLSS 2018. [[slides](#)]
- Deep Generative Models. Aditya Grover and Stefano Ermon. IJCAI 2018. [[slides](#)]
- Planting the Seeds of Probabilistic Thinking: Foundations, Tricks and Algorithms. Shakir Mohamed. MLSS 2018 [[slides](#)][[video](#)]
- Variational Inference: Foundations and Innovations. David M. Blei. [[slides](#)]
- Deep Latent-Variable Models for Natural Language. Yoon Kim, et al. EMNLP 2018. [[link](#)]
- Variational Inference and Deep Generative Models. Wilker Aziz and Philip Schulz. ACL 2018. [[link](#)]

Monographs

- An Introduction to Variational Autoencoders. DP Kingma, Max Welling. Arxiv 2019. [[link](#)]
- Deep Latent Variable Models for Sequential Data. Marco Fraccaro. PhD Thesis 2018. [[link](#)]

References: AR

Image

- [Generating Diverse High-Fidelity Images with VQ-VAE-2](#). Ali Razavi, et al. Arxiv 2019
- [Generating High Fidelity Images with Subscale Pixel Networks and Multidimensional Upscaling](#). Jacob Menick, Nal Kalchbrenner. ICLR 2019
- [Hierarchical Autoregressive Image Models with Auxiliary Decoders](#). JD Fauw, et al. Arxiv 2019
- [PixelSNAIL: An Improved Autoregressive Generative Model](#). XI Chen, et al. ICML 2018
- [Image Transformer](#). Niki Parmar, et al. ICML 2018

Sequence

- [XLNet: Generalized Autoregressive Pretraining for Language Understanding](#). Z Yang, et al. Arxiv 2019
- [MelNet: A Generative Model for Audio in the Frequency Domain](#). S Vasquez, M Lewis. Arxiv 2019
- [Generating Long Sequences with Sparse Transformers](#). Rewon Child, et al. Arxiv 2019
- [Scaling Autoregressive Video Models](#). Dirk Weissenborn, et al. Arxiv 2019
- [Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context](#). Z Dai, et al. ACL 2019
- [Music Transformer: Generating Music with Long-Term Structure](#). CA Huang, et al. ICLR 2019
- [The challenge of realistic music generation: modelling raw audio at scale](#). Sander Dieleman, et al. NeurIPS 2018

Graph

- [GRAM: Scalable Generative Models for Graphs with Graph Attention Mechanism](#). W Kawai, et al. Arxiv 2019
- [Generative Code Modeling with Graphs](#). M Brockschmidt, et al. ICLR 2019
- [GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models](#). J You, et al. ICML 2018

References: LVM (1)

General

- Reweighted Expectation Maximization. Adji B. Dieng, John Paisley. Arxiv 2019
- Doubly Reparameterized Gradient Estimators for Monte Carlo Objectives. George Tucker, et al. ICLR 2019
- A Contrastive Divergence for Combining Variational Inference and MCMC. F Ruiz and MK Titsias. ICML 2019
- MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets. PA Mattei, et al. ICML 2019
- Variational Autoencoder with Arbitrary Conditioning. Oleg Ivanov, et al. ICLR 2019
- DIVA: Domain Invariant Variational Autoencoders. Maximilian Ilse, et al. Arxiv 2019

Image

- Generating Diverse High-Fidelity Images with VQ-VAE-2. Ali Razavi, et al. Arxiv 2019
- BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. Lars Maaløe, et al. Arxiv 2019
- Diagnosing and Enhancing VAE Models. Bin Dai and David Wipf. ICLR 2019
- Multi-Object Representation Learning with Iterative Variational Inference. Klaus Greff, et al. ICML 2019
- A Hierarchical Probabilistic U-Net for Modeling Multi-Scale Ambiguities. Simon Kohl, et al. Arxiv 2019
- Spatial Broadcast Decoder: A Simple Architecture for Learning Disentangled Representations in VAEs. Nicholas Watters, et al. Arxiv 2019

References: LVM (2)

Sequence

- Learning Latent Dynamics for Planning from Pixels. Danijar Hafner, et al. ICML 2019
- Temporal Difference Variational Auto-Encoder. Karol Gregor, et al. ICLR 2019
- Stochastic Latent Actor-Critic: Deep Reinforcement Learning with a Latent Variable Model. Alex X. Lee, et al. Arxiv 2019
- Latent ODEs for Irregularly-Sampled Time Series. Yulia Rubanova, et al. Arxiv 2019
- ODE2VAE: Deep generative second order ODEs with Bayesian neural networks. Çağatay Yıldız, et al. Arxiv 2019
- Neural Ordinary Differential Equations. Ricky T. Q. Chen, et al. NeurIPS 2018
- Effective Estimation of Deep Generative Language Models. Tom Pelsmaeker, Wilker Aziz. Arxiv 2019
- A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. Adam Roberts, et al. ICML 2018
- STCN: Stochastic Temporal Convolutional Networks. Emre Aksan, Otmar Hilliges. ICLR 2019
- Z-Forcing: Training Stochastic Recurrent Networks. Anirudh Goyal, et al. NIPS 2017
- Sequential Neural Models with Stochastic Layers. Marco Fraccaro, et al. NIPS 2016

Graph / Set

- Graphite: Iterative Generative Modeling of Graphs. Aditya Grover, et al. ICML 2019
- Stochastic Blockmodels meet Graph Neural Networks. Nikhil Mehta, et al. ICML 2019
- PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows. Guandao Yang, et al. CVPR 2019
- Junction Tree Variational Autoencoder for Molecular Graph Generation. Wengong Jin, et al. ICML 2018

References: Flow

General

- Neural Spline Flows. Conor Durkan, et al. Arxiv 2019
- Residual Flows for Invertible Generative Modeling. Ricky T. Q. Chen, et al. Arxiv 2019
- FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. Will Grathwohl, et al. ICLR 2019
- Integer Discrete Flows and Lossless Compression. Emiel Hoogeboom, et al. Arxiv 2019
- Discrete Flows: Invertible Generative Models of Discrete Data. Dustin Tran, et al. Arxiv 2019
- AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows. Aditya Grover, et al. Arxiv 2019

Image

- Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. Jonathan Ho, et al. ICML 2019
- Glow: Generative Flow with Invertible 1x1 Convolutions. Diederik P. Kingma, Prafulla Dhariwal. NeurIPS 2018

Sequence

- Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. Joan Serrà, et al. Arxiv 2019
- Latent Normalizing Flows for Discrete Sequences. Zachary M. Ziegler and Alexander M. Rush. ICML 2019
- FloWaveNet: A Generative Flow for Raw Audio. Sungwon Kim, et al. ICML 2019
- VideoFlow: A Flow-Based Generative Model for Video. Manoj Kumar, et al. Arxiv 2019

Graph

- Graph Normalizing Flows. Jenny Liu, et al. Arxiv 2019
- GraphNVP: An Invertible Flow Model for Generating Molecular Graphs. Kaushalya Madhawa, et al. Arxiv 2019

References

OOD Detection

- WAIC, but Why? Generative Ensembles for Robust Anomaly Detection. Hyunsun Choi, et al. Arxiv 2019
- Likelihood Ratios for Out-of-Distribution Detection. Jie Ren, et al. Arxiv 2019
- Detecting Out-of-Distribution Inputs to Deep Generative Models Using a Test for Typicality. Eric Nalisnick, et al. Arxiv 2019
- Conditional Generative Models are not Robust. Ethan Fetaya, et al. Arxiv 2019
- Normalizing flows for novelty detection in industrial time series data. M Schmidt and M Simic. ICML 2019 INNF Workshop
- AdaFlow: Domain-Adaptive Density Estimator with Application to Anomaly Detection and Unpaired Cross-Domain Translation. Masataka Yamaguchi, et al. ICASSP 2019
- Anomaly Detection in Raw Audio Using Deep Autoregressive Networks. E Rushe, et al. ICASSP 2019

Applications in DB

- Selectivity Estimation with Deep Likelihood Models. Zongheng Yang, et al. Arxiv 2019
- Approximate Query Processing for Data Exploration using Deep Generative Models. S Thirumuruganathan, et al. Arxiv 2019

Thanks!

Questions?