

# Machine learning for physicists

<https://github.com/wangleiphy/ml4p>

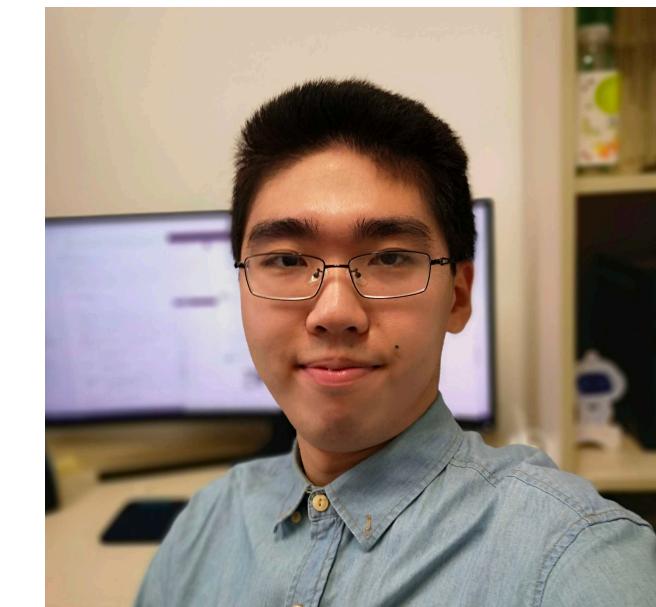
每周四上午10点

课程微信群

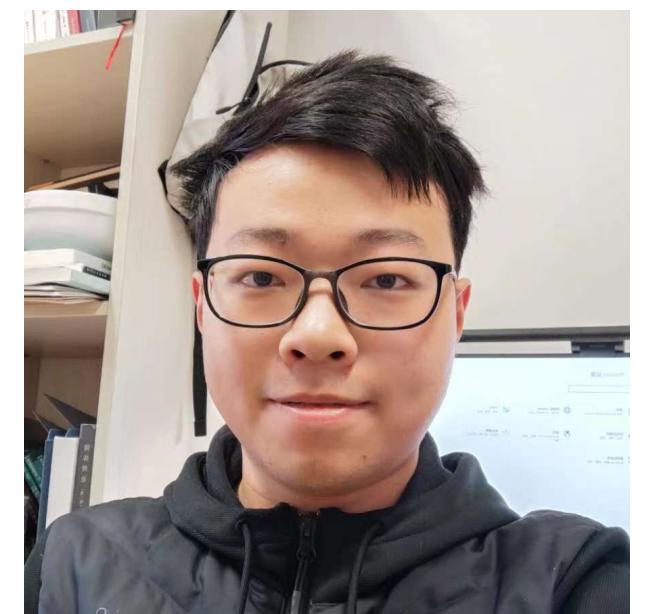
2.23	Overview
3.2	Machine learning practices
3.9	A hitchhiker's guide to deep learning
3.16	Research projects hands-on
3.23	Symmetries in machine learning
3.30	Differentiable programming
4.6	Generative models-I
4.13	Generative models-II
4.20	Research projects presentation
4.27	AI for science: why now ?



助教



李子航



李扬帆

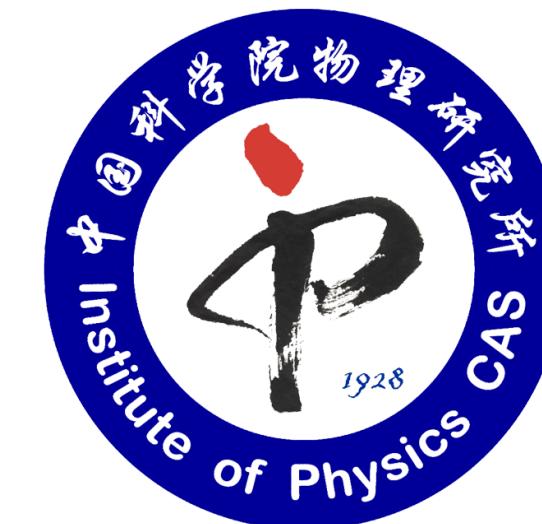
考核方式: project + presentation (1学分)

# Generative AI for Science

Lei Wang (王磊)

Institute of Physics, CAS

<https://wangleiphy.github.io>



# Science is more than fitting, so is machine learning

## Discriminative learning



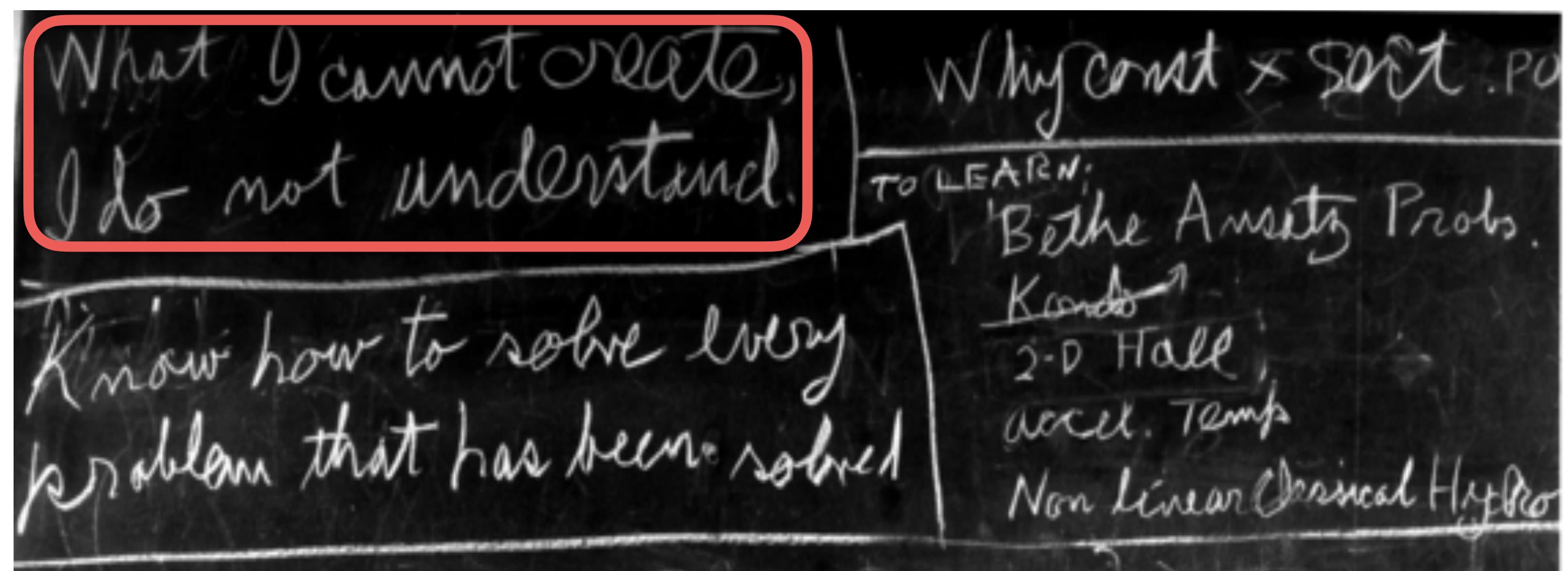
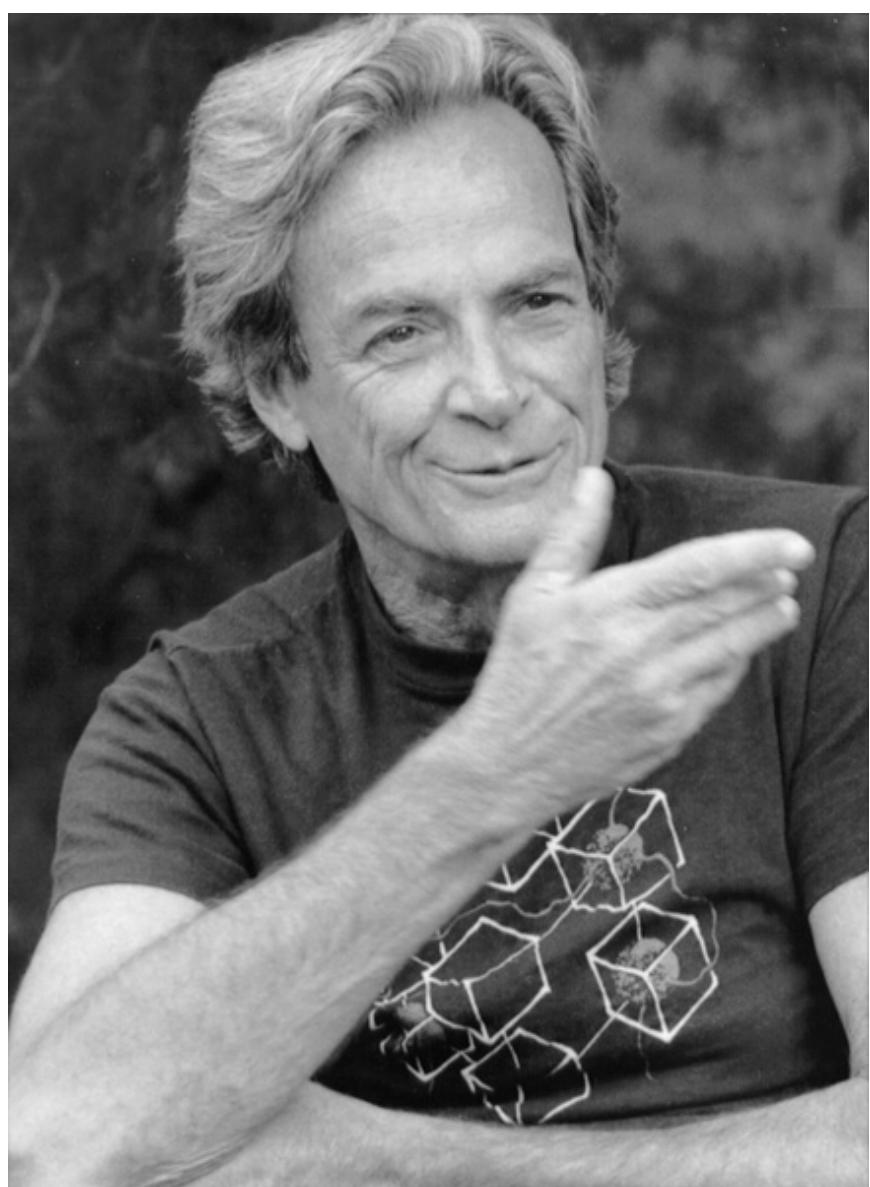
## Generative learning



$$y = f(x)$$

or  $p(y | x)$

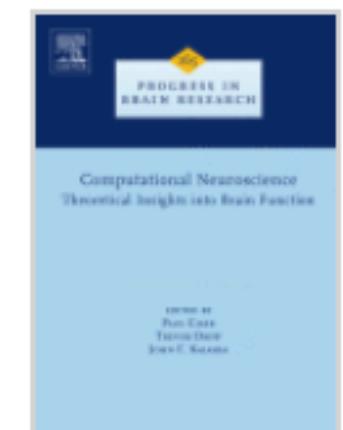
$$p(x, y)$$



## Progress in Brain Research

Volume 165, 2007, Pages 535–547

Computational Neuroscience: Theoretical Insights into Brain Function



To recognize shapes, first learn to generate images

Geoffrey E. Hinton

Department of Computer Science, University of Toronto, 10 Kings College Road, Toronto, M5S 3G4  
Canada

**ChatGPT: Optimizing Language Models for Dialogue**  
November 30, 2022 — Announcements, Research

**DALL·E API Now Available in Public Beta**  
November 3, 2022 — Announcements, API

**DALL·E Now Available Without Waitlist**  
September 28, 2022 — Announcements

**Introducing Whisper**  
September 21, 2022 — Research

**DALL·E: Introducing Outpainting**  
August 31, 2022 — Announcements

**Our Approach to Alignment Research**  
August 24, 2022 — Research

**New and Improved Content Moderation Tooling**  
August 10, 2022 — Announcements

**DALL·E Now Available in Beta**  
July 20, 2022 — Announcements

**OpenAI Technical Goals**  
June 20, 2016 — Announcements

**Generative Models**  
June 16, 2016 — Research, Milestones

**Team Update**  
May 25, 2016 — Announcements

**OpenAI Gym Beta**  
April 27, 2016 — Research

**Welcome, Pieter and Shivon!**  
April 26, 2016 — Announcements

**Team++**  
March 31, 2016 — Announcements

**Introducing OpenAI**  
December 11, 2015 — Announcements

<https://openai.com/blog/>

# Generative AI: a new buzz word in silicon valley

## *A Coming-Out Party for Generative A.I., Silicon Valley's New Craze*

A celebration for Stability AI, the start-up behind the controversial Stable Diffusion image generator, represents the arrival of a new A.I. boom.

New York Times

Kevin Roose

Oct. 21, 2022

Protocol  
Biz Carson  
October 21, 2022

## **Sequoia's Sonya Huang: The generative AI hype is 'absolutely justified'**

She's bullish on generative AI given the "superpowers" it gives humans who work with it.

<https://www.sequoiacap.com/article/generative-ai-a-creative-new-world/>

by Sonya Huang, Pat Grady and GPT-3

	PRE - 2020	2020	2022	2023?	2025?	2030?
TEXT	Spam detection Translation Basic Q&A	Basic copy writing First drafts	Longer form Second drafts	Vertical fine tuning gets good (scientific papers, etc)	Final drafts better than the human average	Final drafts better than professional writers
CODE	1-line auto-complete	Multi-line generation	Longer form Better accuracy	More languages More verticals	Text to product (draft)	Text to product (final), better than full-time developers
IMAGES			Art Logos Photography	Mock-ups (product design, architecture, etc.)	Final drafts (product design, architecture, etc.)	Final drafts better than professional artists, designers, photographers
VIDEO / 3D / GAMING			First attempts at 3D/video models	Basic / first draft videos and 3D files	Second drafts	AI Roblox Video games and movies are personalized dreams

Large model availability:

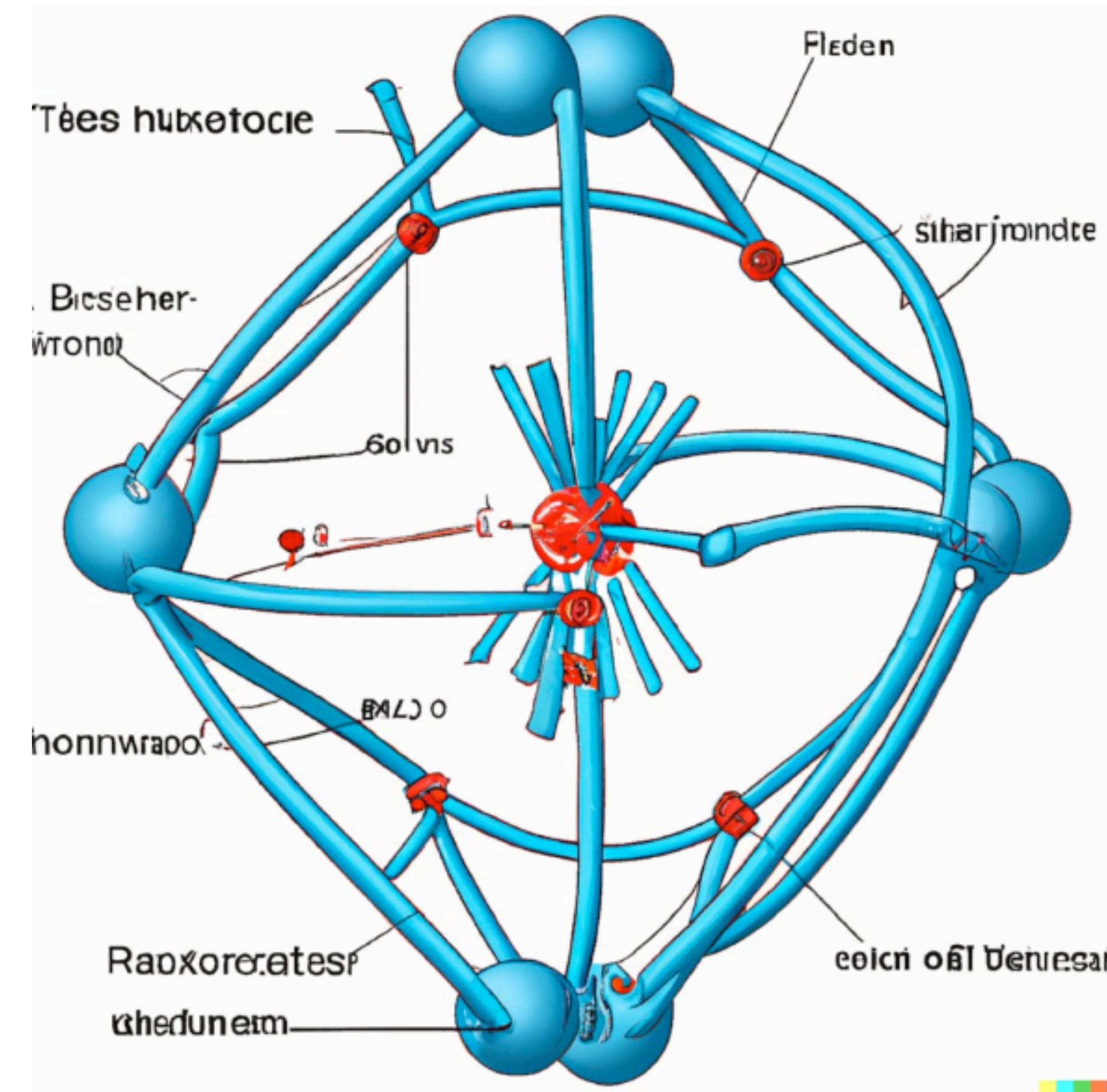
● First attempts

● Almost there

● Ready for prime time

the inner structure of an electron

Generate image



<https://future.com/how-to-build-gpt-3-for-science/>

# How to Build a GPT-3 for Science

(scientific literature and data)

Josh Nicholson

Posted August 18, 2022

Generative Pre-Training

You may ask (prompts):

“Tell me why this hypothesis is wrong”

“Tell me why my treatment idea won’t work”

“Generate a new treatment idea”

“What evidence is there to support social policy X?”

“Who has published the most reliable research in this field?”

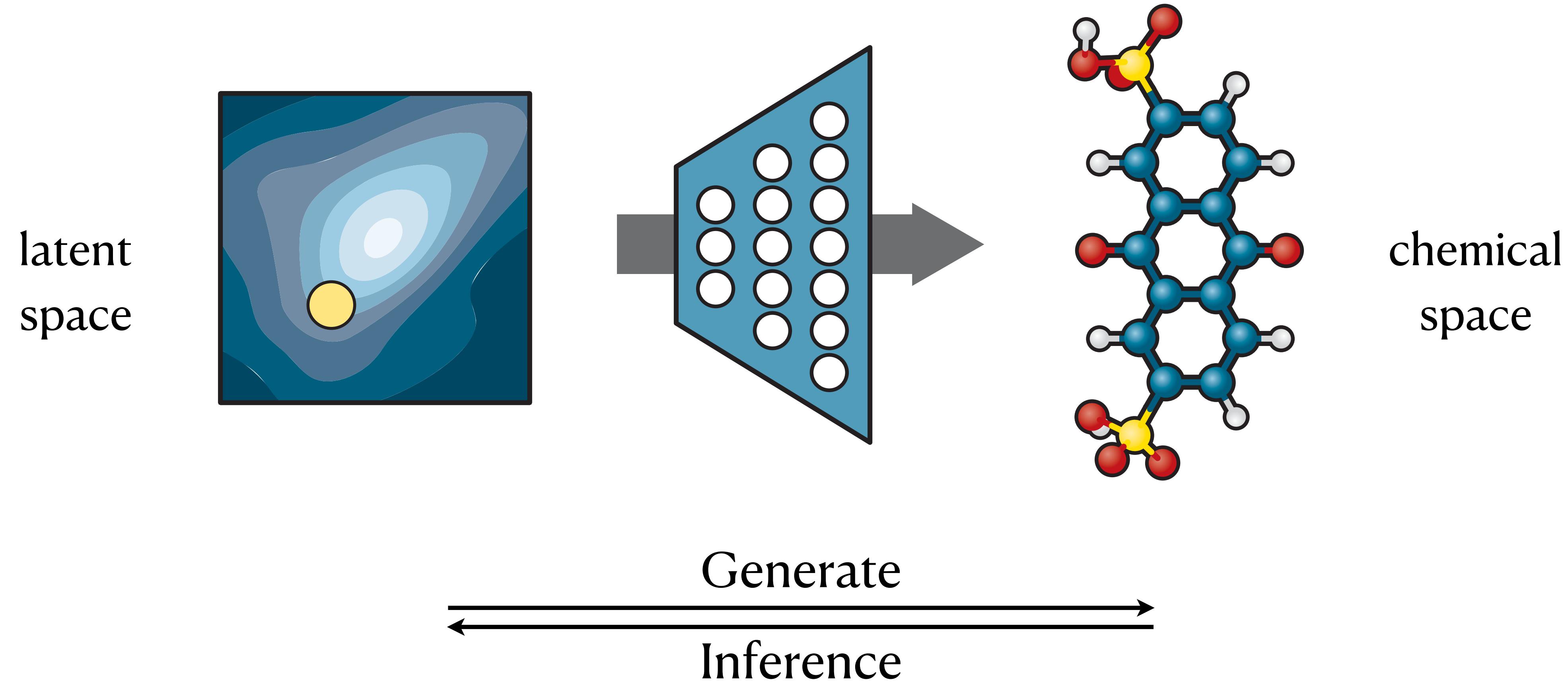
“Write me a scientific paper based on my data”

<https://galactica.org/>

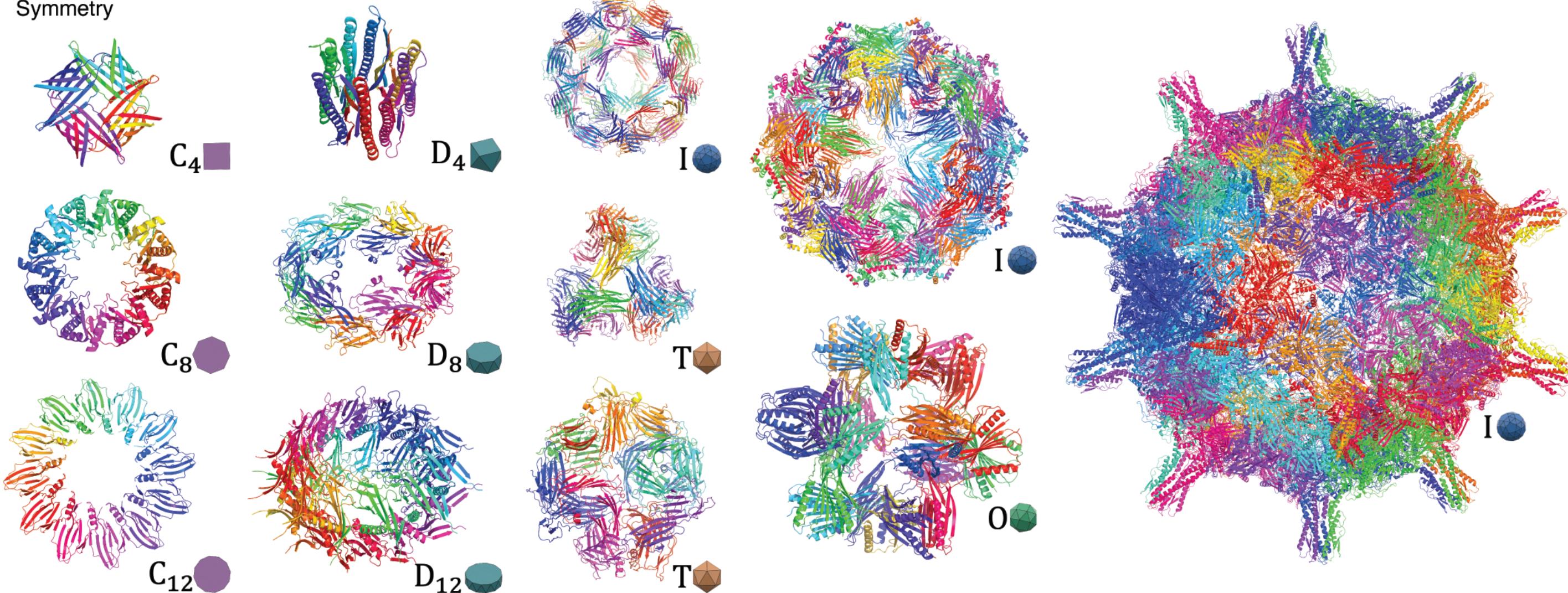
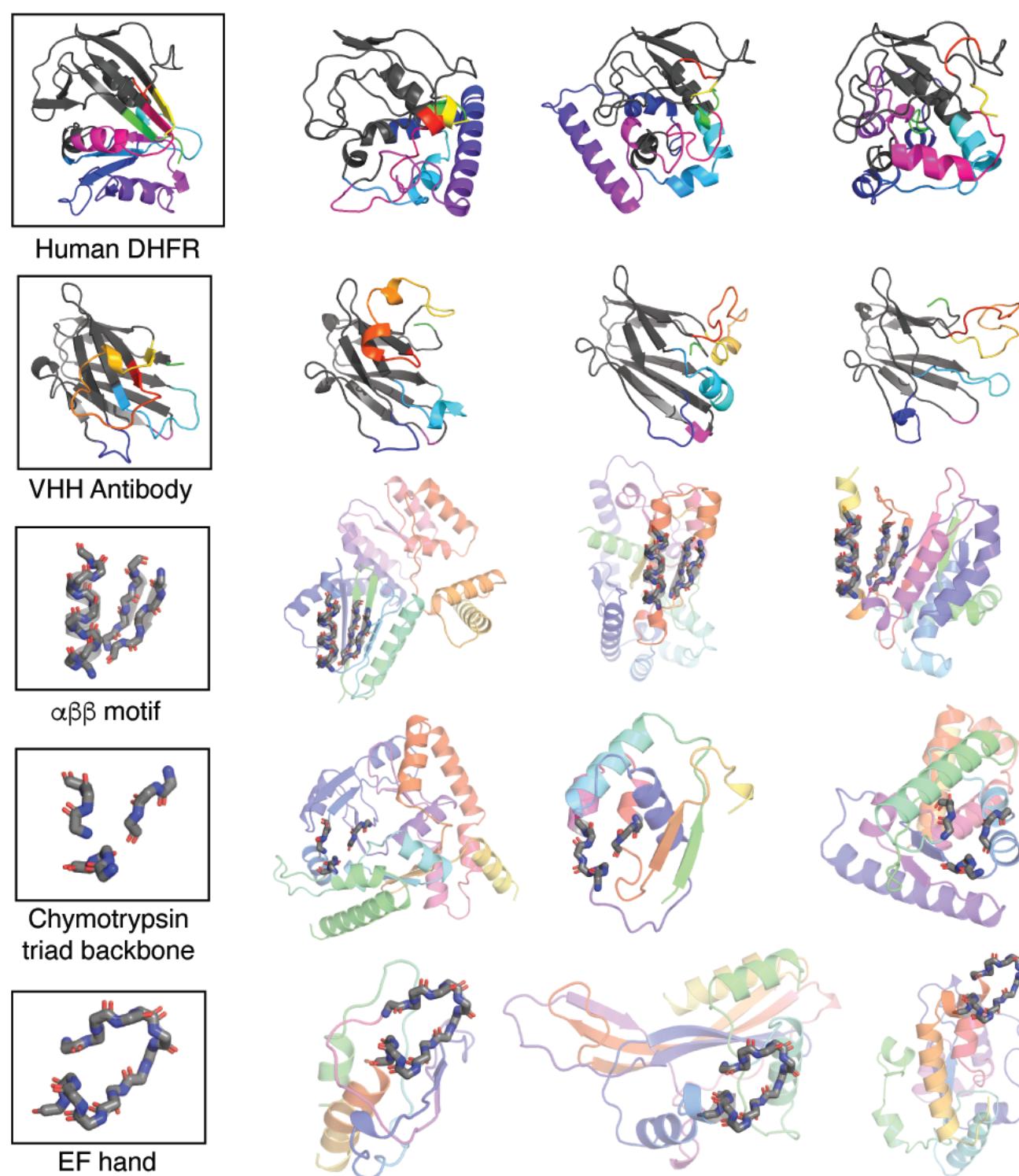
Galactica: A Large Language Model for Science

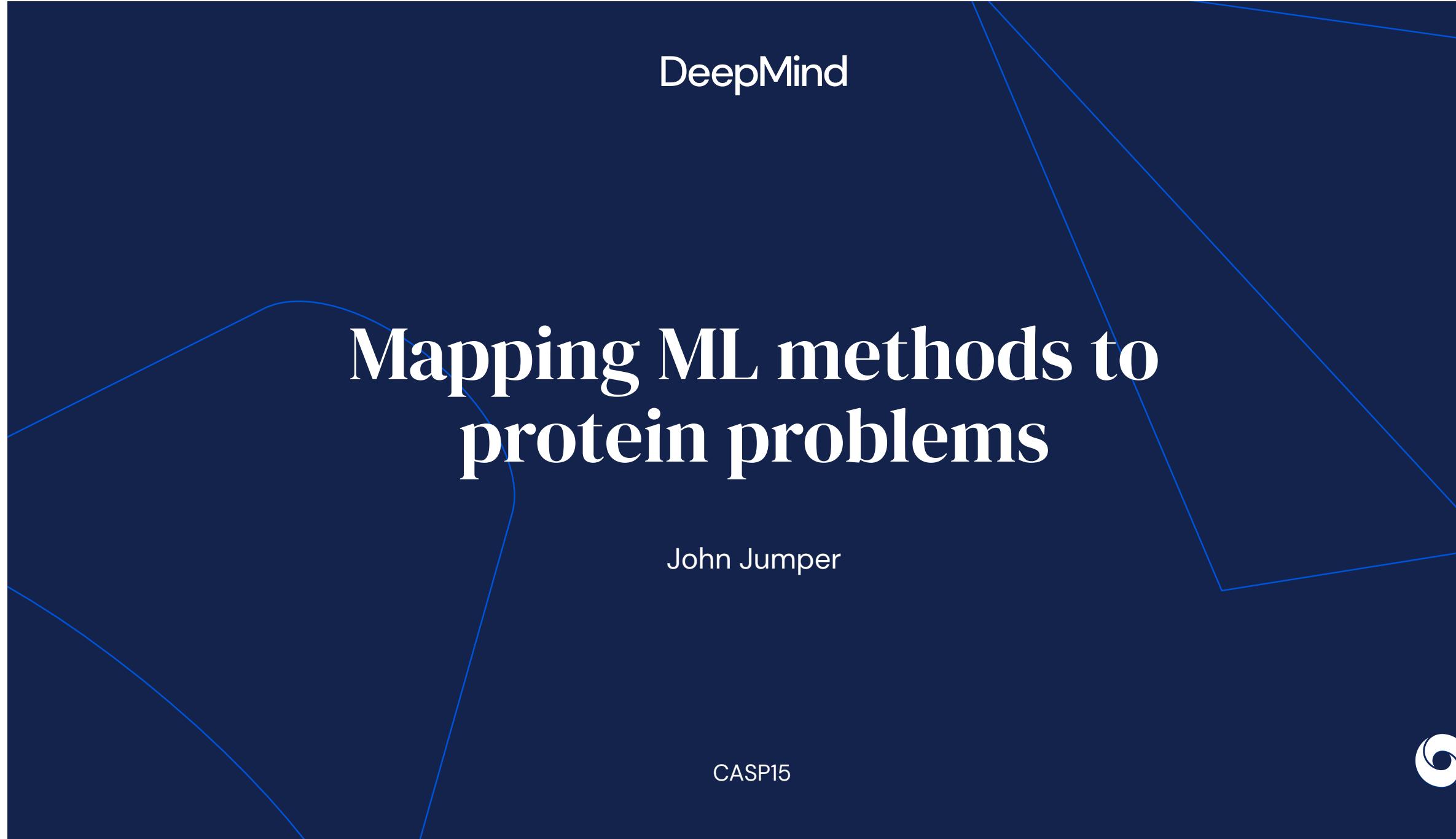
Taylor et al (Meta), 2211.09085

# Generative AI for matter engineering



Review: “Inverse molecular design using machine learning”, Sanchez-Lengeling & Aspuru-Guzik, Science ’18

**a Symmetry****b Substructure****c Shape** $p(\text{protein} \mid \text{symmetry})$  $p(\text{protein} \mid \text{substructure})$  $p(\text{protein} \mid \text{shape})$

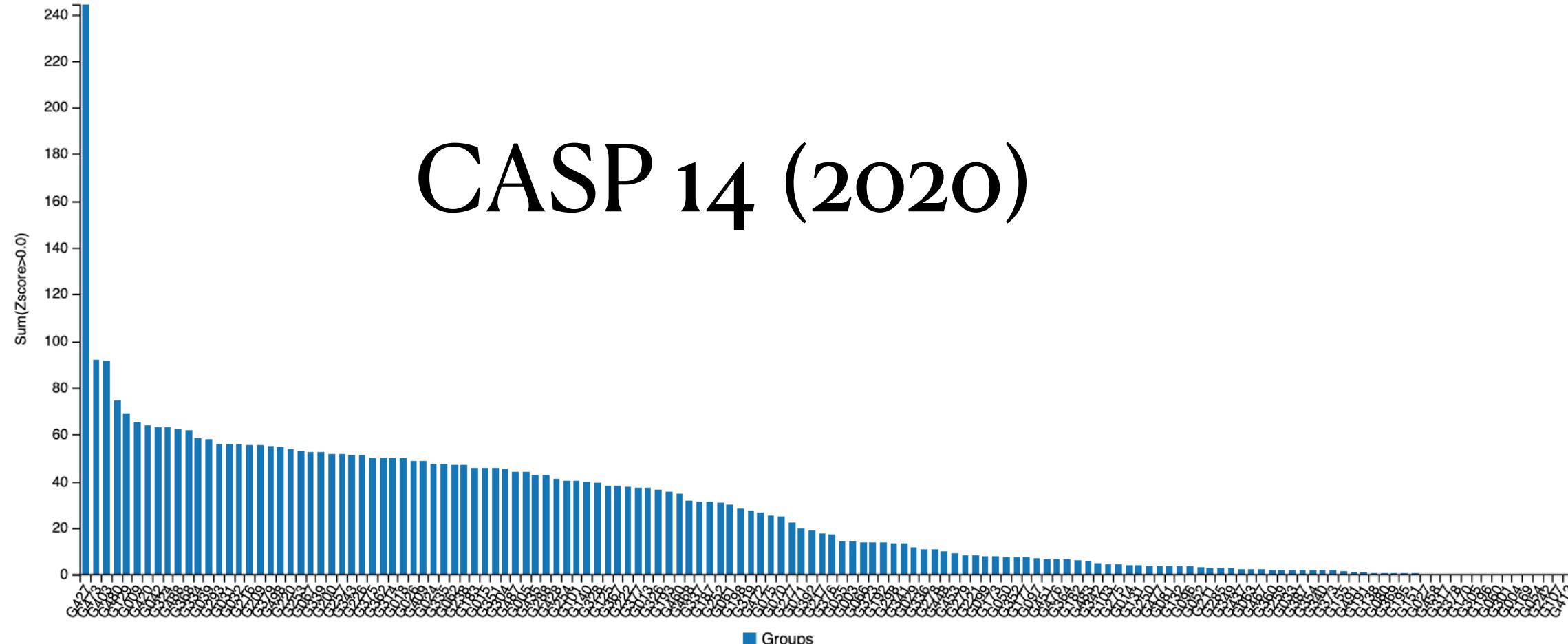


# CASP 15 invited talk by John Jumper

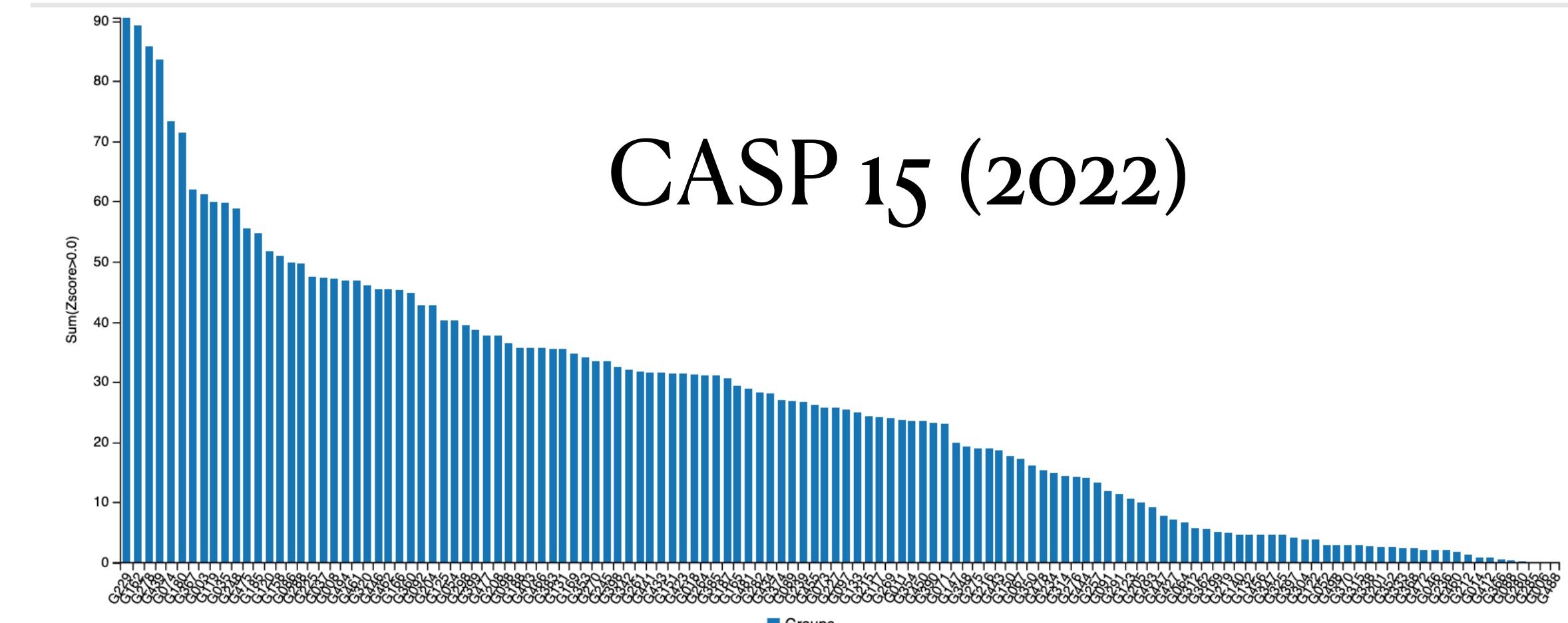
## Outline

- Generative models and diffusion
- Protein language models and the scaling hypothesis
- Next problems

CASP 14 (2020)



CASP 15 (2022)

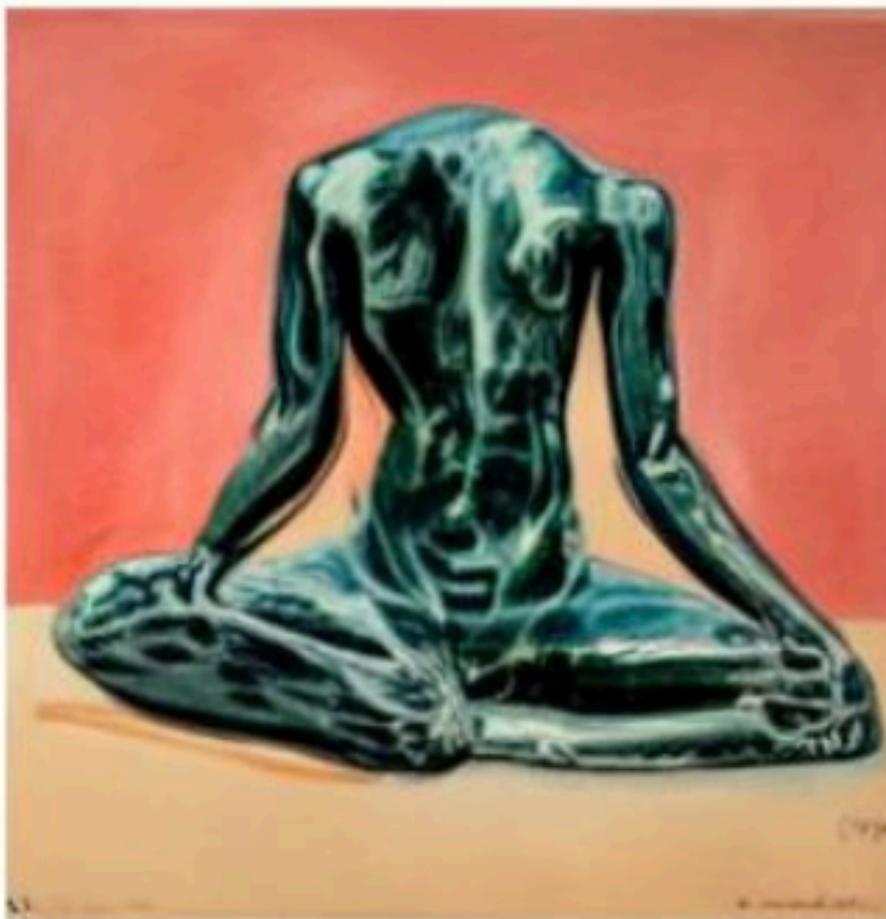


# Great to have for protein generative models

## textual inversion



→



Input samples  $\xrightarrow{\text{invert}}$  “ $S_*$ ”

“An oil painting of  $S_*$ ”

“App icon of  $S_*$ ”

“Elmo sitting in  
the same pose as  $S_*$ ”

“Crochet  $S_*$ ”

# Great to have for protein generative models

## Instruct-pix2pix

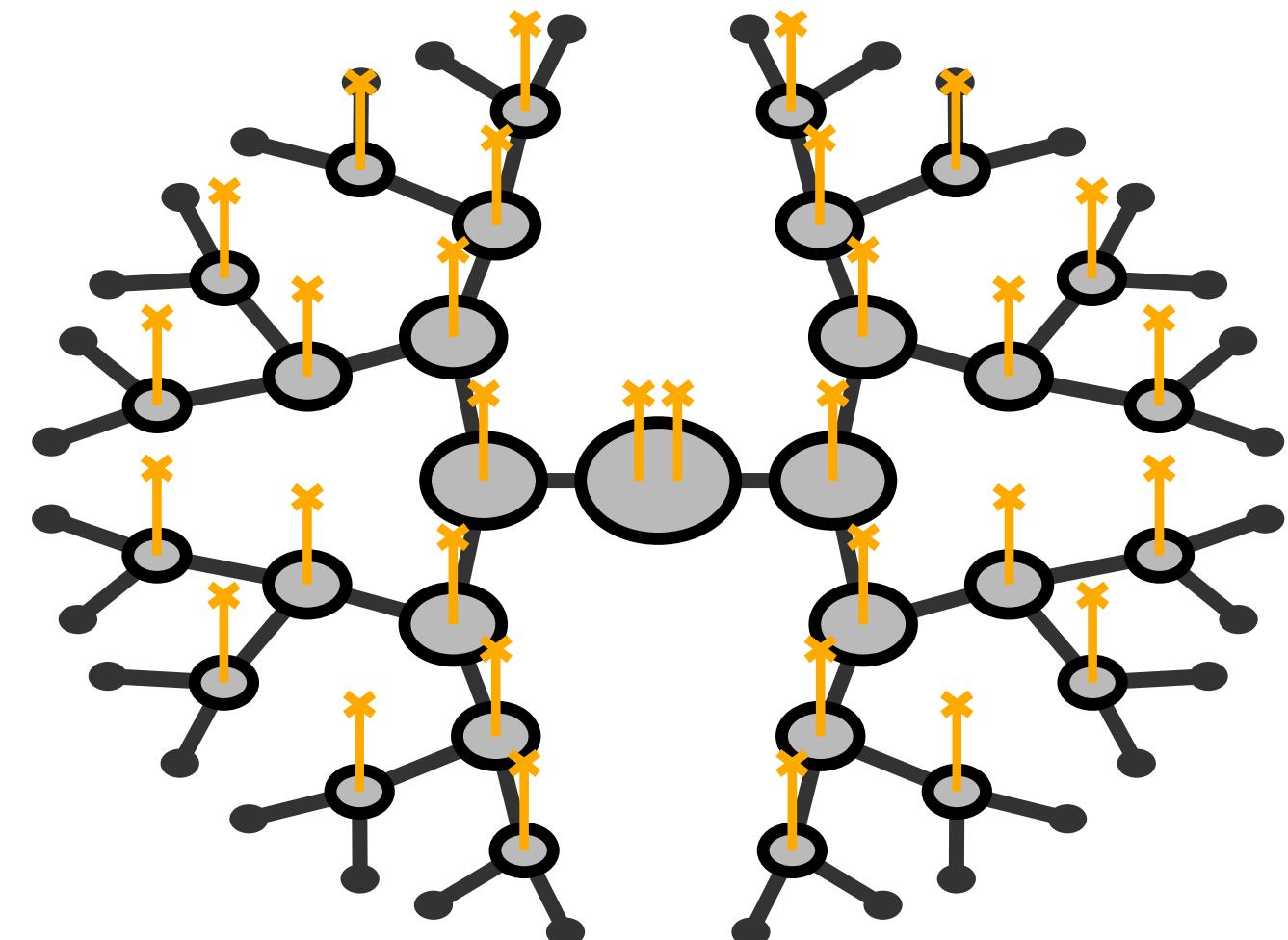


Given an image and a written instruction, our method follows the instruction to edit the image.

<https://www.timothybrooks.com/instruct-pix2pix>

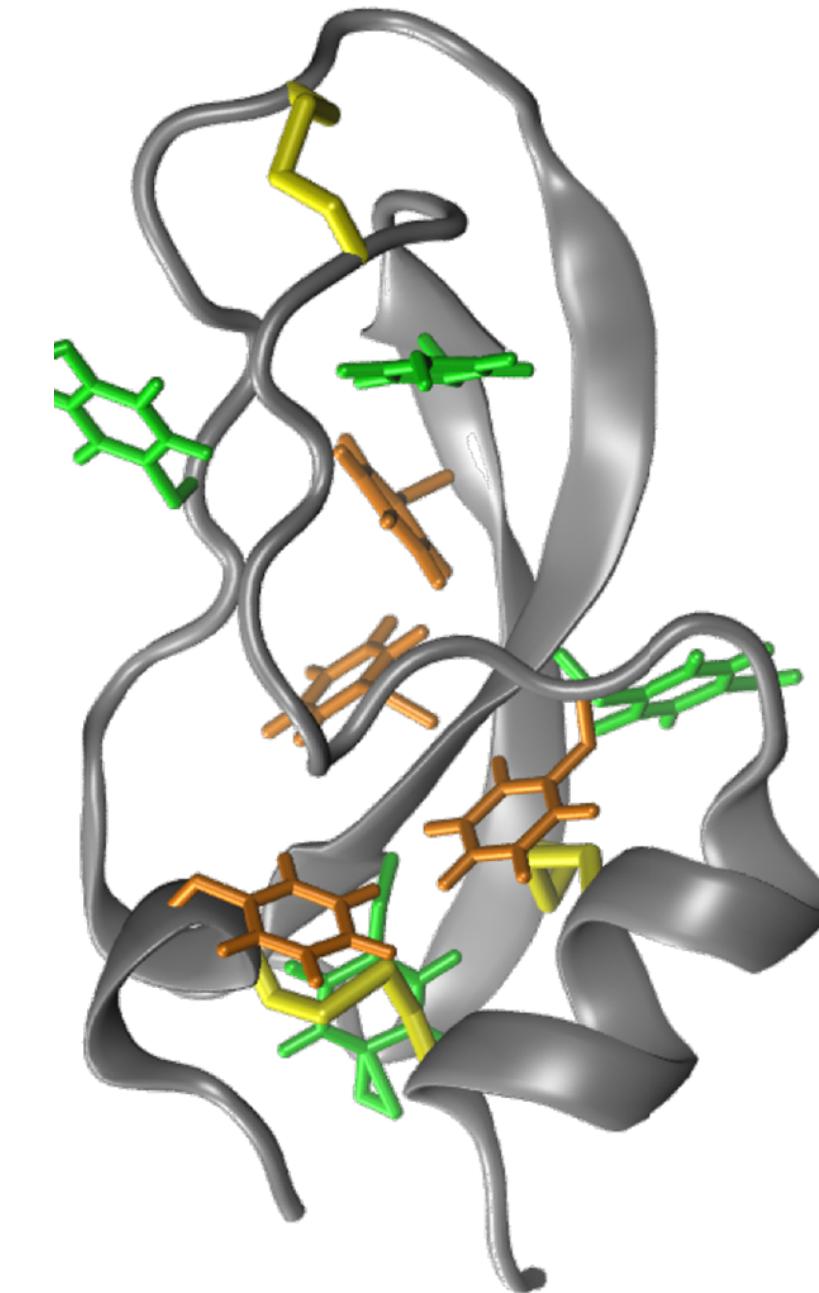
# Generative AI for matter computation

## Renormalization group



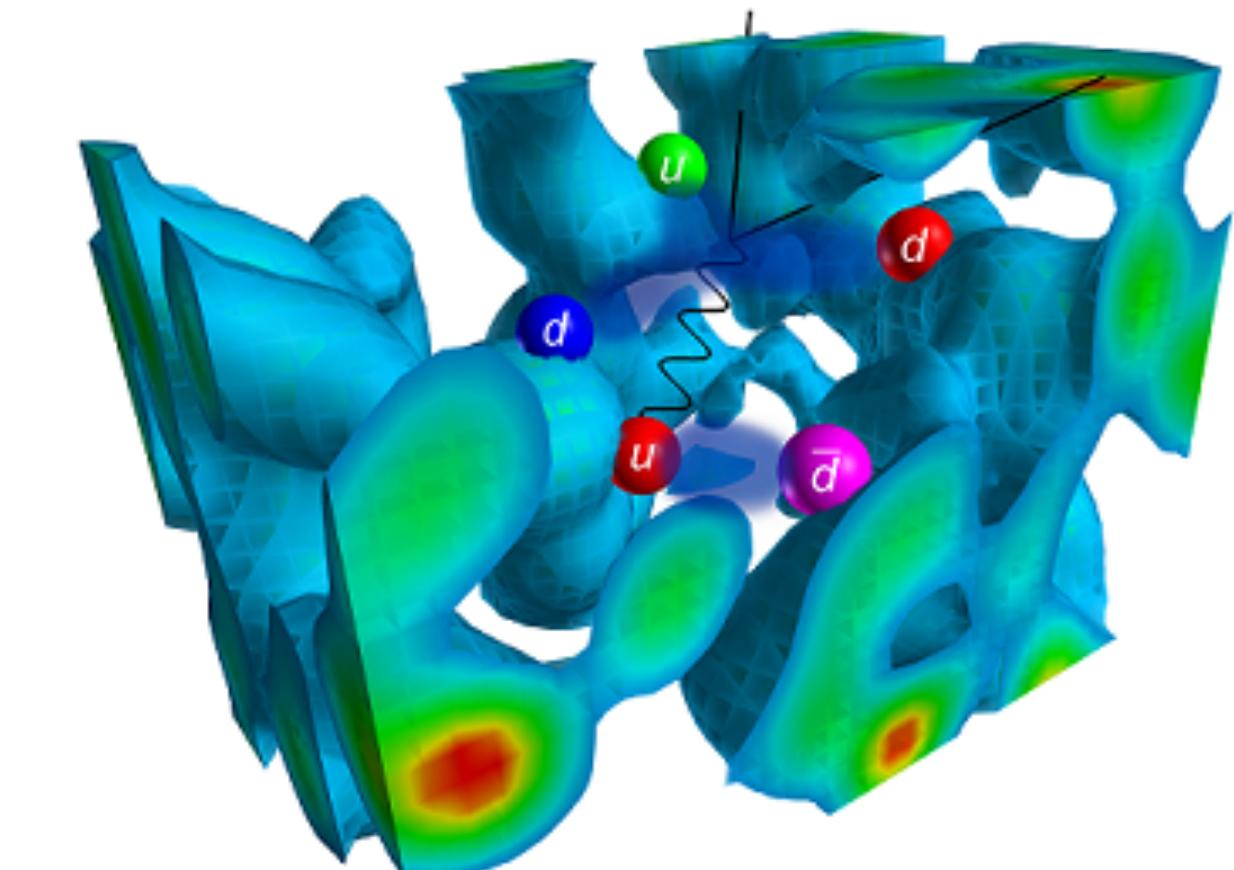
Li and LW, PRL '18  
Li, Dong, Zhang, LW, PRX '20

## Molecular simulation



Noe et al, Science '19  
Wirnsberger et al, JCP '20

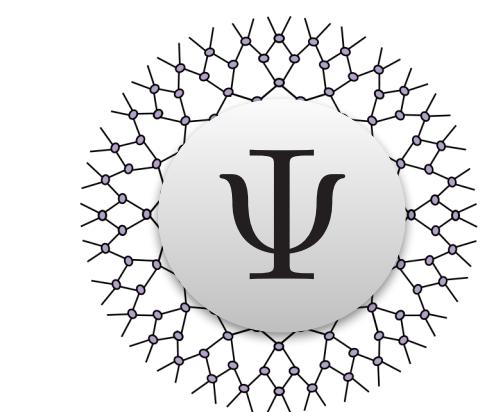
## Lattice field theory



Albergo et al, PRD '19  
Kanwar et al, PRL '20

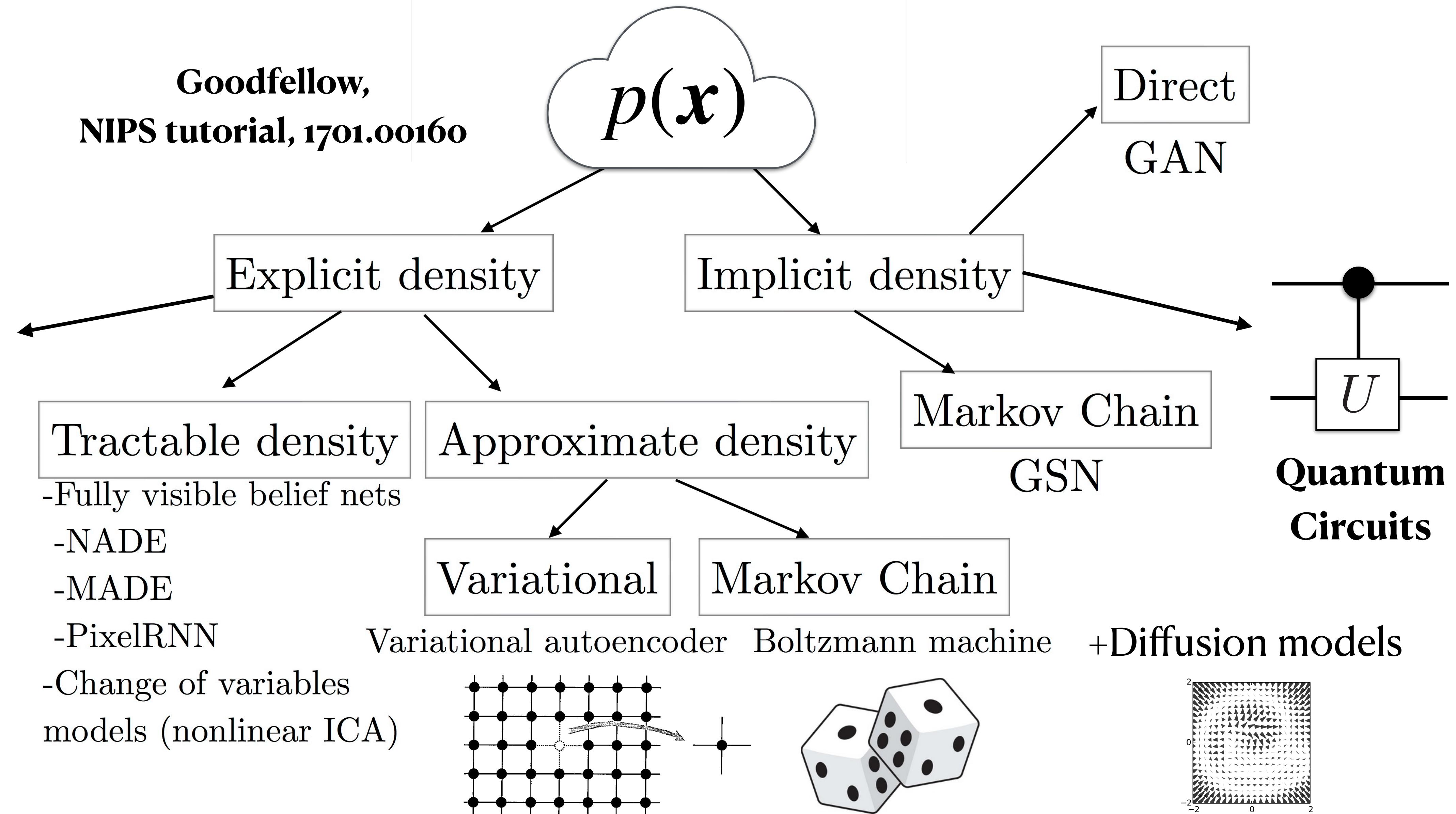
These are principled computation: quantitatively accurate,  
interpretable, reliable, and generalizable even without data

# Generative models and their physics genes

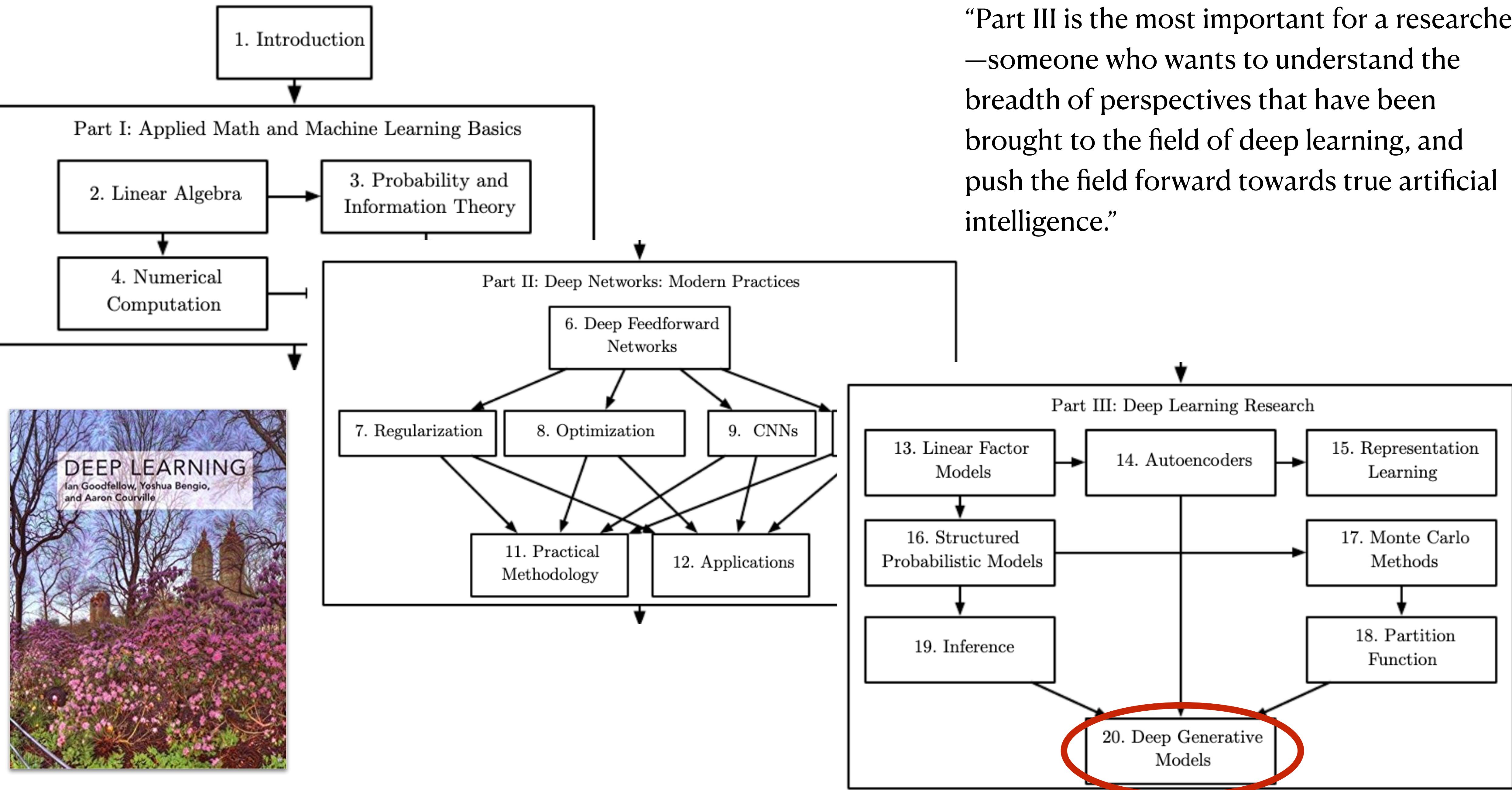


**Tensor  
Networks**

**Goodfellow,  
NIPS tutorial, 1701.00160**



“Part III is the most important for a researcher—someone who wants to understand the breadth of perspectives that have been brought to the field of deep learning, and push the field forward towards true artificial intelligence.”



# Generative Models for Physicists

Lei Wang\*

Institute of Physics, Chinese Academy of Sciences  
Beijing 100190, China

October 28, 2018

## Abstract

Generative models generate unseen samples according to a learned joint probability distribution in the high-dimensional space. They find wide applications in density estimation, variational inference, representation learning and more. Deep generative models and associated techniques (such as differentiable programming and representation learning) are cutting-edge technologies physicists can learn from deep learning.

This note introduces the concept and principles of generative modeling, together with applications of modern generative models (autoregressive models, normalizing flows, variational autoencoders etc) as well as the old ones (Boltzmann machines) to physics problems. As a bonus, this note puts some emphasize on physics-inspired generative models which take insights from statistical, quantum, and fluid mechanics.

The latest version of the note is at <http://wangleiphy.github.io/>. Please send comments, suggestions and corrections to the email address in below.

## CONTENTS

---

<b>1 GENERATIVE MODELING</b>	<b>2</b>
1.1 Probabilistic Generative Modeling	2
1.2 Generative Model Zoo	4
1.2.1 Boltzmann Machines	5
1.2.2 Autoregressive Models	8
1.2.3 Normalizing Flow	9
1.2.4 Variational Autoencoders	13
1.2.5 Tensor Networks	15
1.2.6 Generative Adversarial Networks	17
1.2.7 Generative Moment Matching Networks	18
1.3 Summary	20
<b>2 PHYSICS APPLICATIONS</b>	<b>21</b>
2.1 Variational Ansatz	21
2.2 Renormalization Group	22
2.3 Monte Carlo Update Proposals	22
2.4 Chemical and Material Design	23
2.5 Quantum Information Science and Beyond	24
<b>3 RESOURCES</b>	<b>25</b>
<b>BIBLIOGRAPHY</b>	<b>26</b>

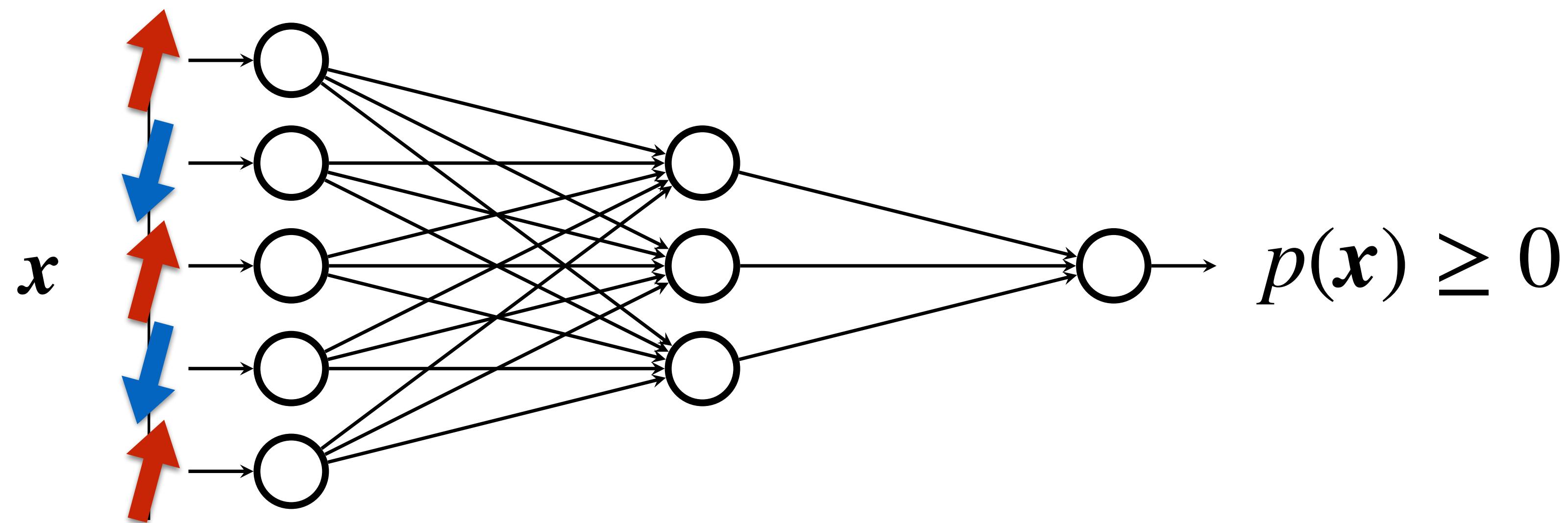
\*I will update the note with materials contained in this lecture once I find time

# (outdated\*) lecture note <http://wangleiphy.github.io/lectures/PILtutorial.pdf>

Table 1: A summary of generative models and their salient features. Question marks mean generalizations are possible, but nontrivial.

Name	Training Cost	Data Space	Latent Space	Architecture	Sampling	Likelihood	Expressibility	Difficulty (Learn/Sample)
RBM	Log-likelihood	Arbitrary	Arbitrary	Bipartite	MCMC	Intractable partition function	★	💀/💀
DBM	ELBO	Arbitrary	Arbitrary	Bipartite	MCMC	Intractable partition function & posterior	★★★	💀/💀
Autoregressive Model	Log-likelihood	Arbitrary	None	Ordering	Sequential	Tractable	★★	💀/💀
Normalizing Flow	Log-likelihood	Continuous	Continuous, Same dimension as data	Bijector	Parallel	Tractable	★★	💀/💀
VAE	ELBO	Arbitrary	Continuous	Arbitrary?	Parallel	Intractable posterior	★★★	💀/💀
MPS/TTN	Log-likelihood	Arbitrary?	None or tree tensor	No loop	Sequential	Tractable	★★★	💀/💀
GAN Quantum Circuit	Adversarial Adversarial	Continuous Discrete	Arbitrary? Discrete	Arbitrary Arbitrary	Parallel Parallel	Implicit Implicit	★★★★ ★★★★★	💀/💀

# So, what is the fuss ?



**Normalization ?**

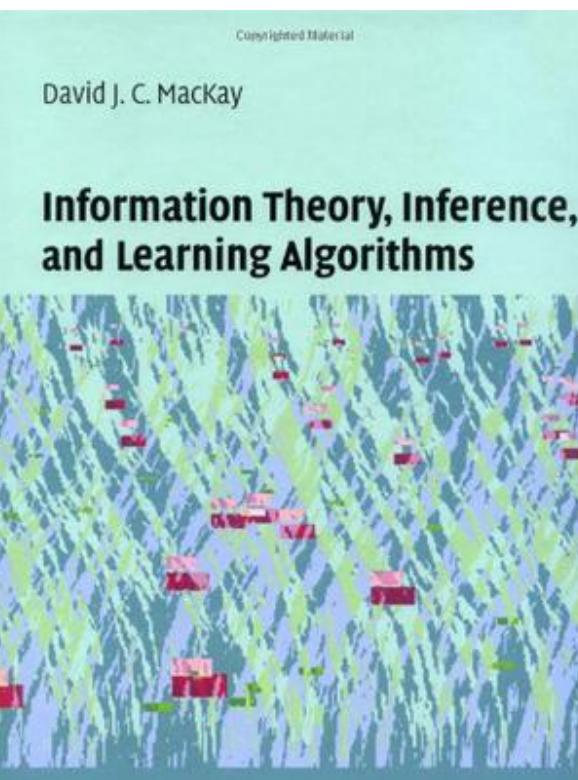
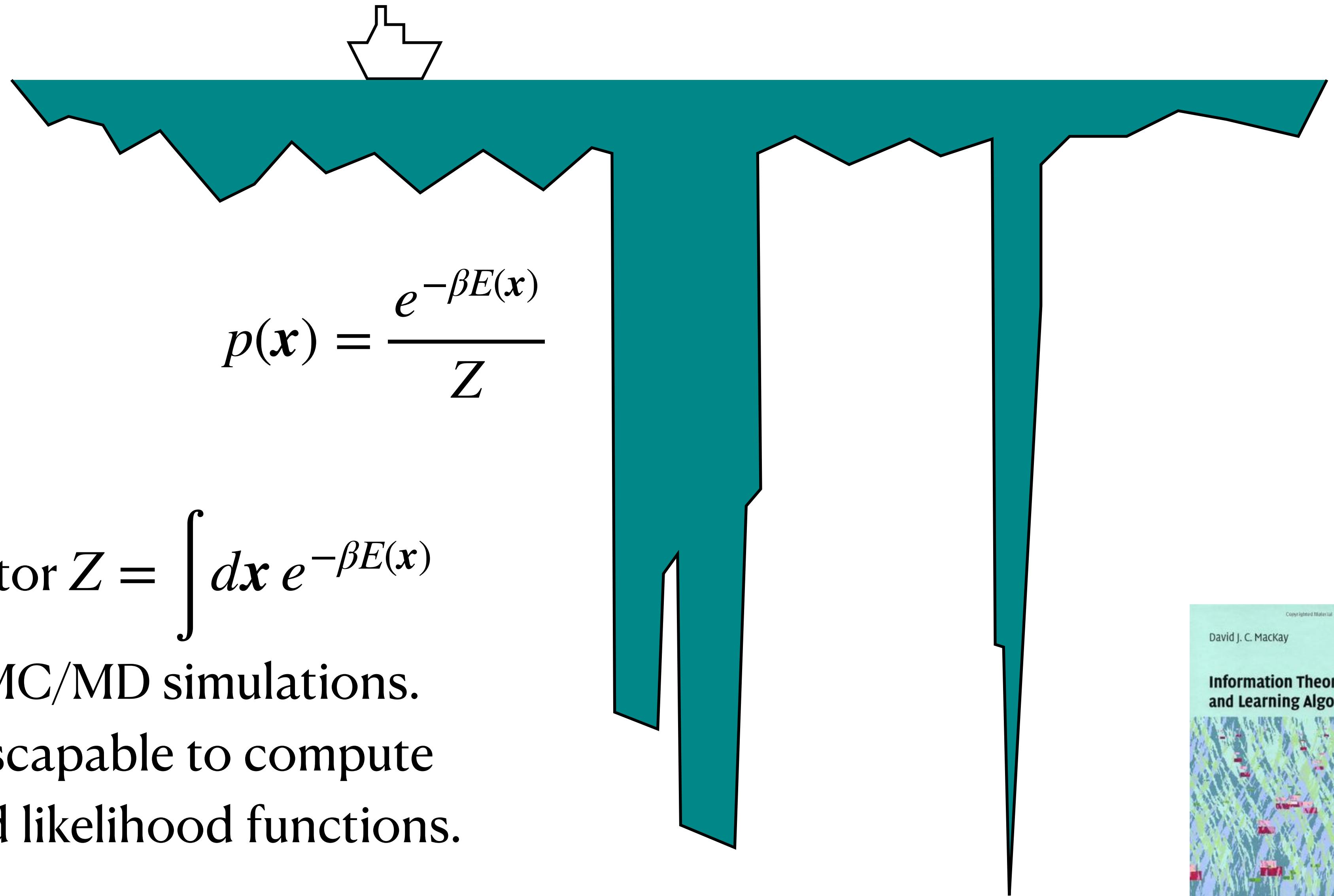
$$\int dx p(x)$$

**Sampling ?**

$$\mathbb{E}_{x \sim p(x)}$$

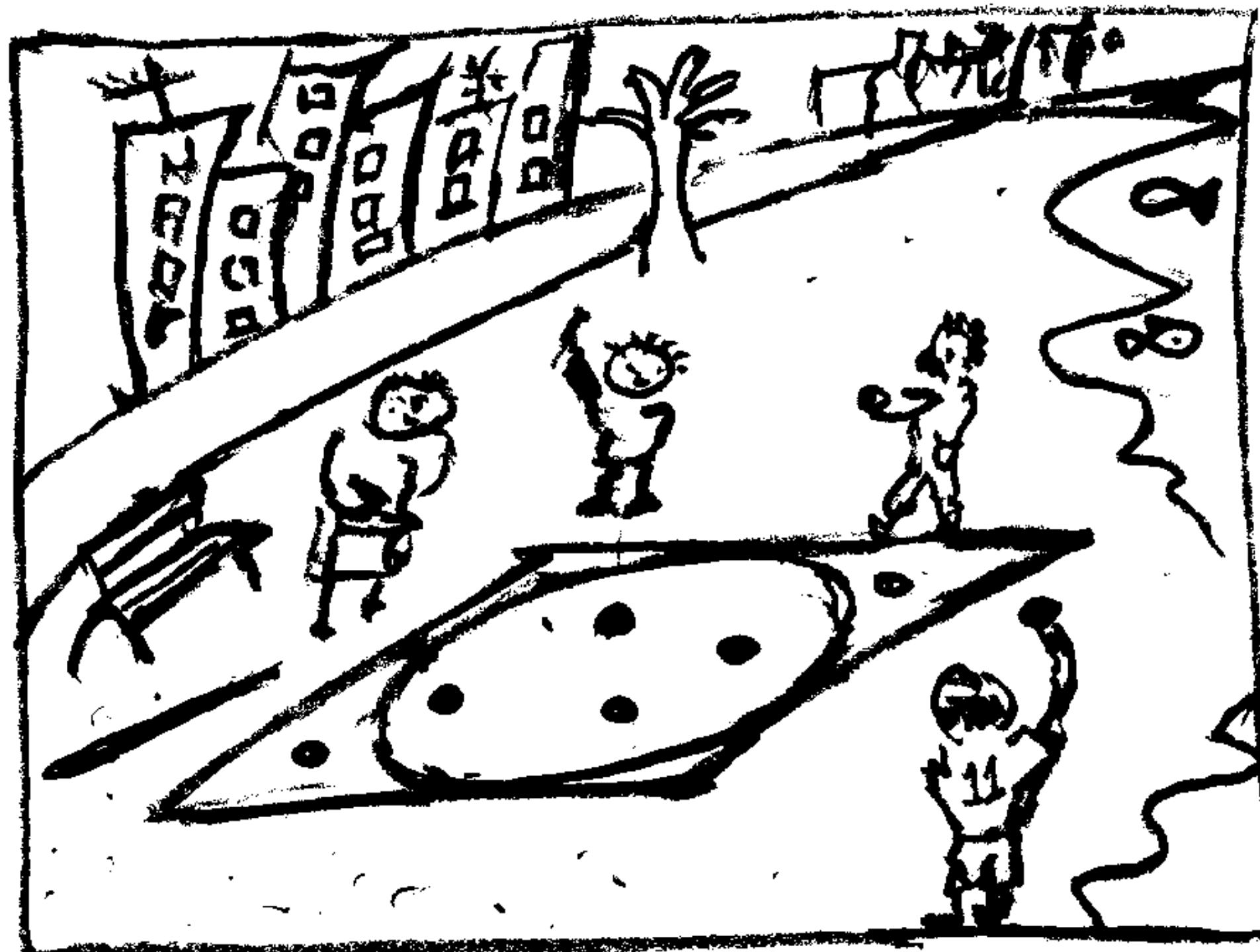
# Normalization

The normalization factor  $Z = \int dx e^{-\beta E(x)}$   
does not show up in MC/MD simulations.  
However, they are inescapable to compute  
entropy, free energy and likelihood functions.



# Direct sampling

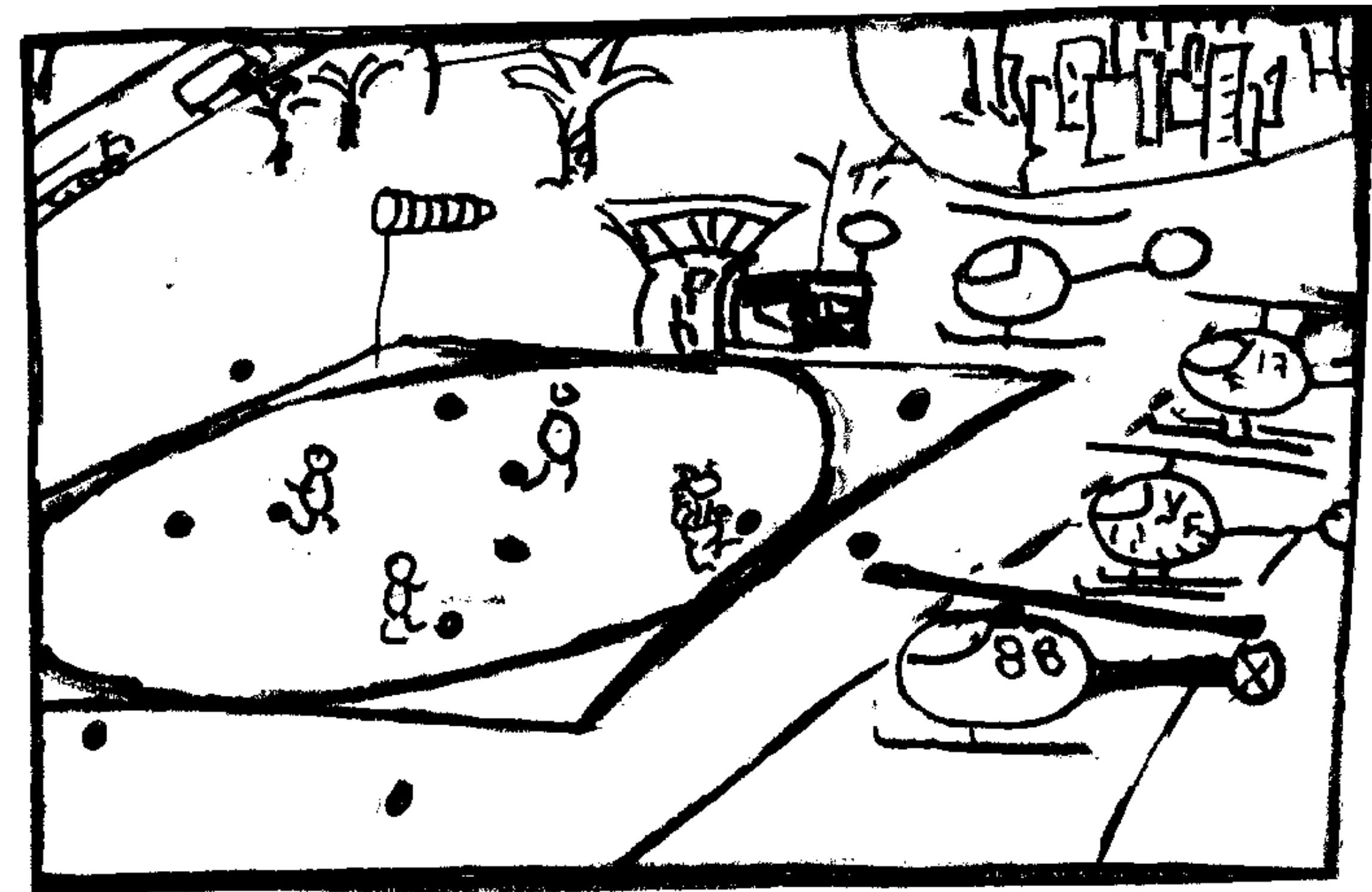
“perfect”: uncorrelated samples



Children computing the number  $\pi$  on the Monte Carlo beach.

# Markov chain Monte Carlo

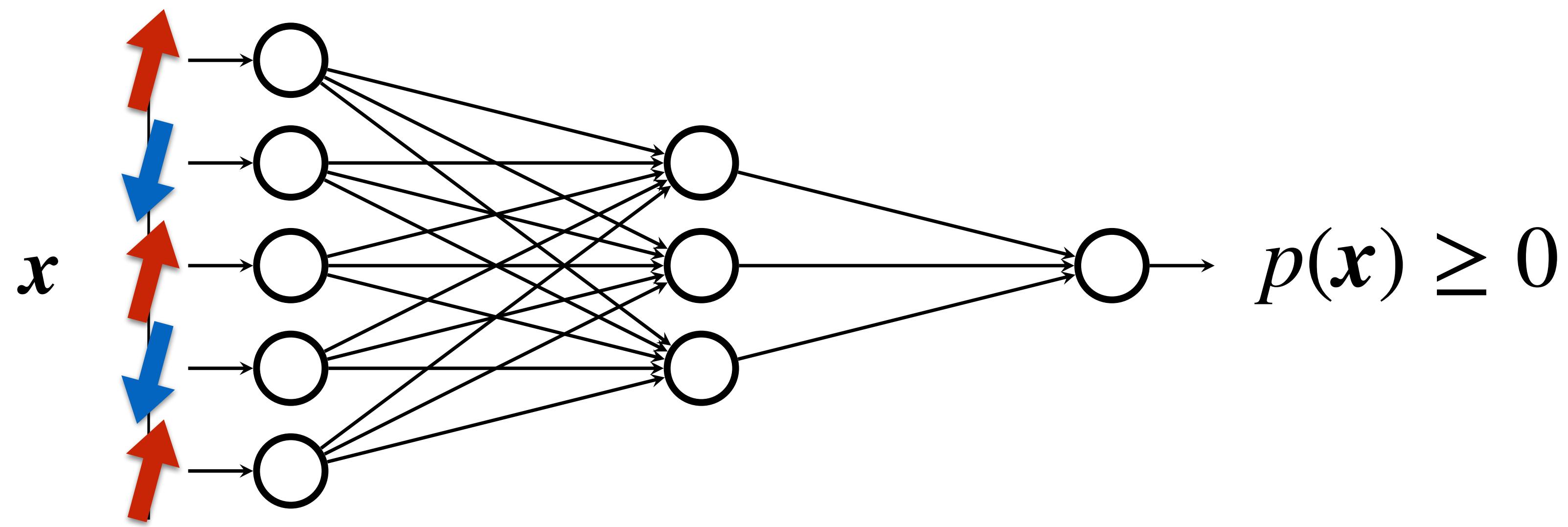
“imperfect”: correlated samples



Adults computing the number  $\pi$  at the Monte Carlo heliport.



# So, what is the fuss ?



**Normalization ?**

$$\int dx p(x)$$

We are going to see several powerful generative models that are normalized and/or samples directly!

**Sampling ?**

$$\mathbb{E}_{x \sim p(x)}$$

## Generative models

Negative log-likelihood

Score function

Latent variables

Partition function

Sample diversity

## Statistical physics

Energy function

Force

Collective variables/coarse  
graining/renormalization group

Free energy calculation

Enhanced sampling

# Two sides of the same coin

## Generative modeling



Known: samples

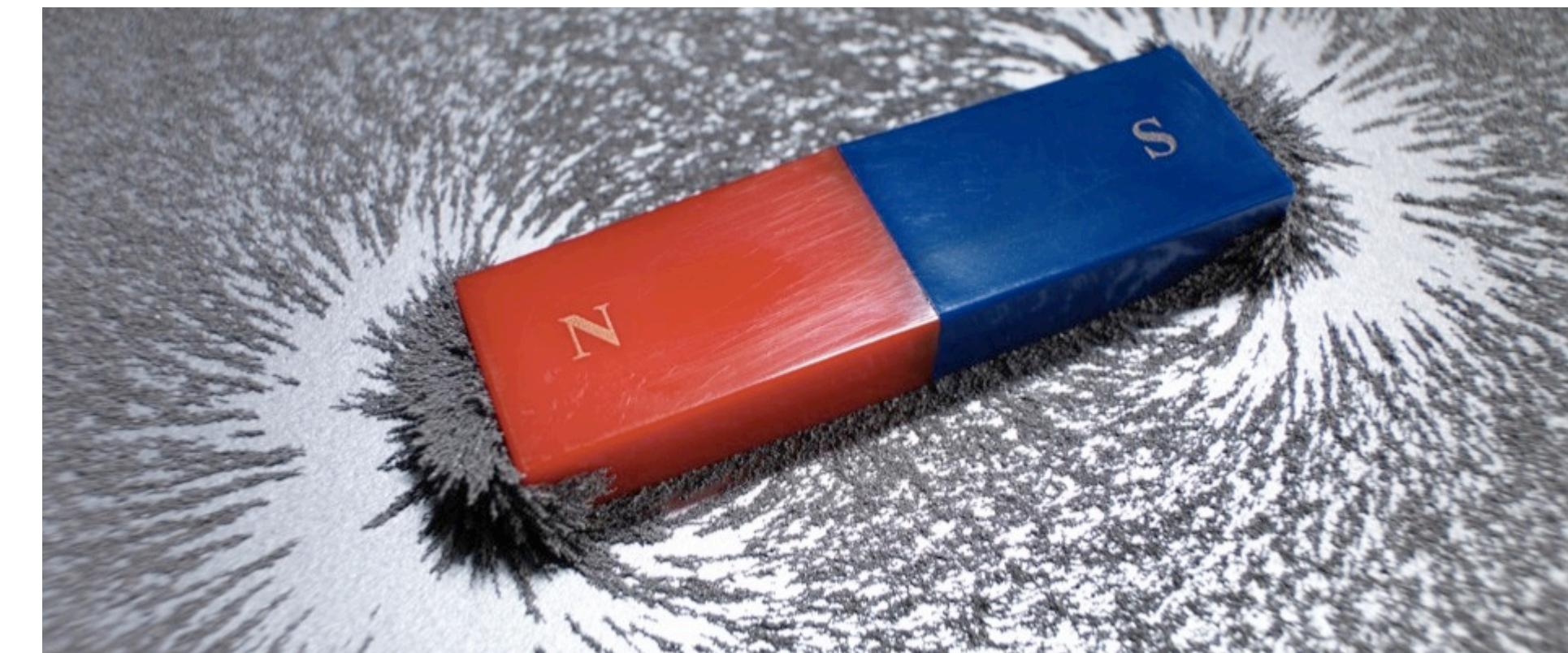
Unknown: generating distribution

**“learn from data”**

$$\mathcal{L} = - \mathbb{E}_{x \sim \text{data}} [\ln p(x)]$$

$$\mathbb{K}\mathbb{L}(\text{data} \parallel p) \text{ vs } \mathbb{K}\mathbb{L}(p \parallel e^{-H/k_B T})$$

## Statistical physics



Known: energy function

Unknown: samples, partition function

**“learn from Hamiltonian”**

$$F = \mathbb{E}_{x \sim p(x)} [H(x) + k_B T \ln p(x)]$$

# Kullback–Leibler divergence

$$\mathbb{KL}(\pi \parallel p) \equiv \int dx \pi(x) [\ln \pi(x) - \ln p(x)]$$

$$\mathbb{KL}(\pi \parallel p) \geq 0$$

$$\mathbb{KL}(\pi \parallel p) = 0 \iff \pi(x) = p(x)$$

$$\mathbb{KL}(\pi \parallel p) \neq \mathbb{KL}(p \parallel \pi)$$

# Learn from data

$$\pi(x) \propto \sum_{d \in \text{data}} \delta(x - d)$$

$$\min_{\theta} \mathbb{KL}(\pi \parallel p_{\theta}) \iff \min_{\theta} \left\{ -\mathbb{E}_{x \sim \text{data}} [\ln p_{\theta}(x)] \right\}$$

↑      ↑  
target    model

Maximum likelihood estimation

# Learn from Hamiltonian

$$\pi(x) \propto e^{-H/k_B T}$$

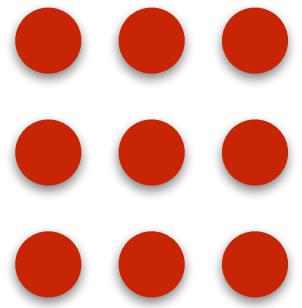
$$\min_{\theta} \text{KL}(p_{\theta} \parallel \pi) \iff \min_{\theta} \left\{ \mathbb{E}_{x \sim p_{\theta}(x)} [H(x) + k_B T \ln p_{\theta}(x)] \right\}$$

↑      ↑  
model      target      Variational free energy

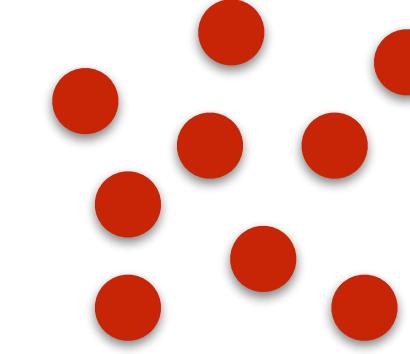
# Nature tries to minimize free energy

$$F = E - TS$$

energy



entropy



$F$  is a **cost function** given by Nature

# The variational free energy principle

$$F[p] = \int dx p(x) [H(x) + k_B T \ln p(x)] \geq F$$

↓                      ↓                      ↓  
variational density    energy    entropy    😱

**Difficulties in Applying the Variational  
Principle to Quantum Field Theories<sup>1</sup>**

Richard P. Feynman

Generative  
models!

<sup>1</sup>transcript of his talk in 1987

# Deep variational free energy approach

Use deep generative models as the variational density

$$F[p] = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [H(\mathbf{x}) + k_B T \ln p(\mathbf{x})]$$

↓                                    ↓  
energy                                 entropy 😊

Li and LW, PRL '18  
Wu, LW, Zhang, PRL '19

with normalizing flow &  
autoregressive models



Tractable entropy



Direct sampling

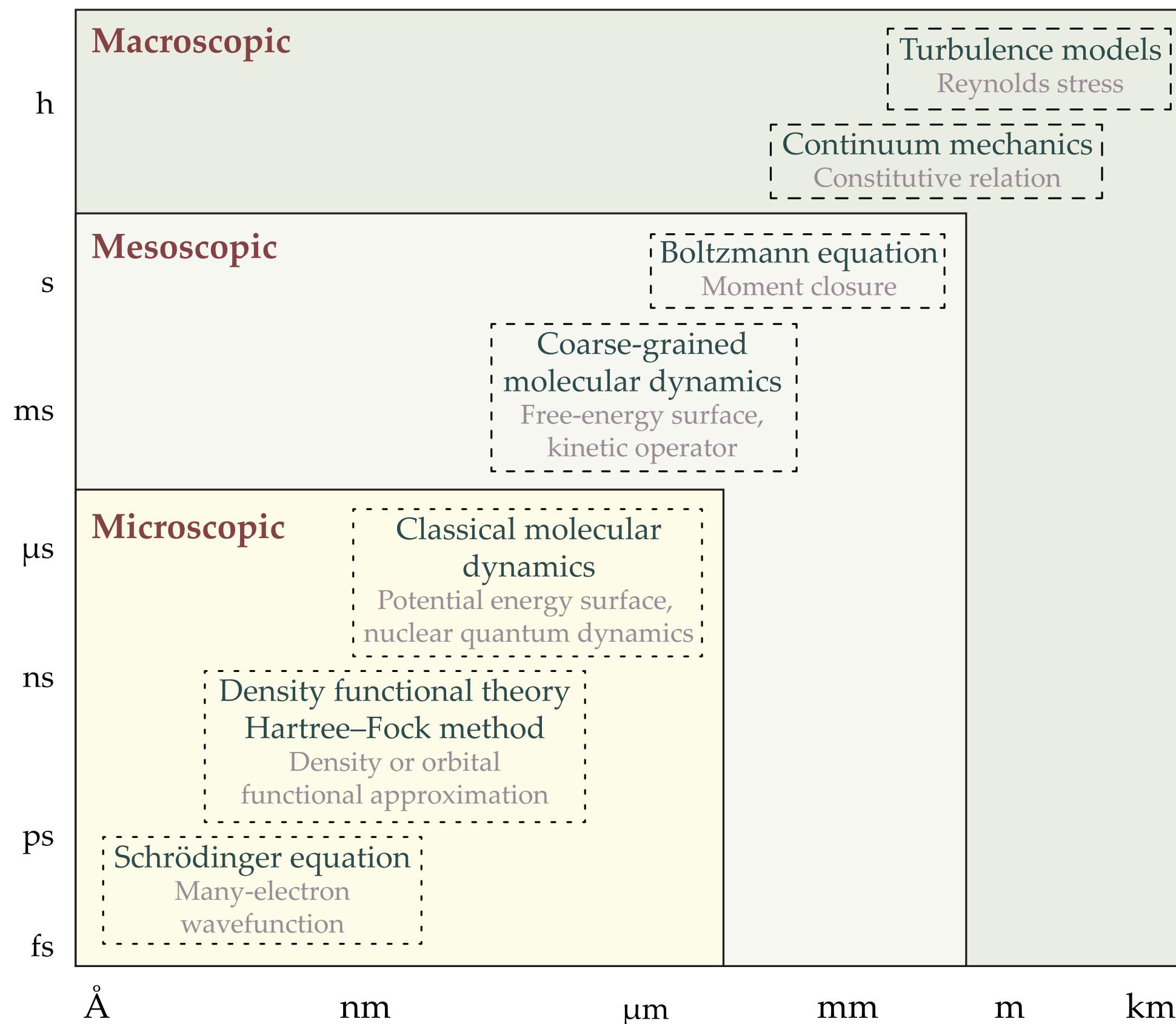


Turning a sampling problem to an optimization problem  
better leverages the deep learning engine:



# Deep variational free-energy in the context

E, Han,Zhang, Physics Today 2020



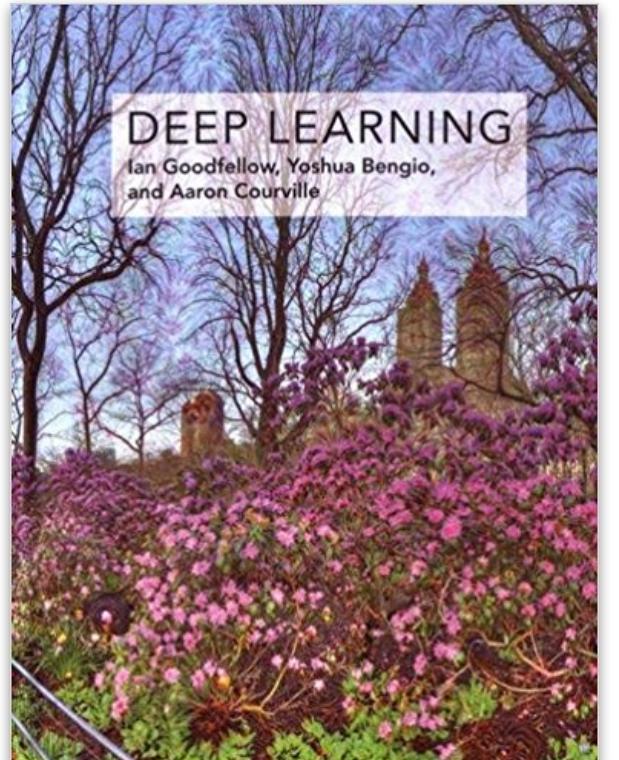
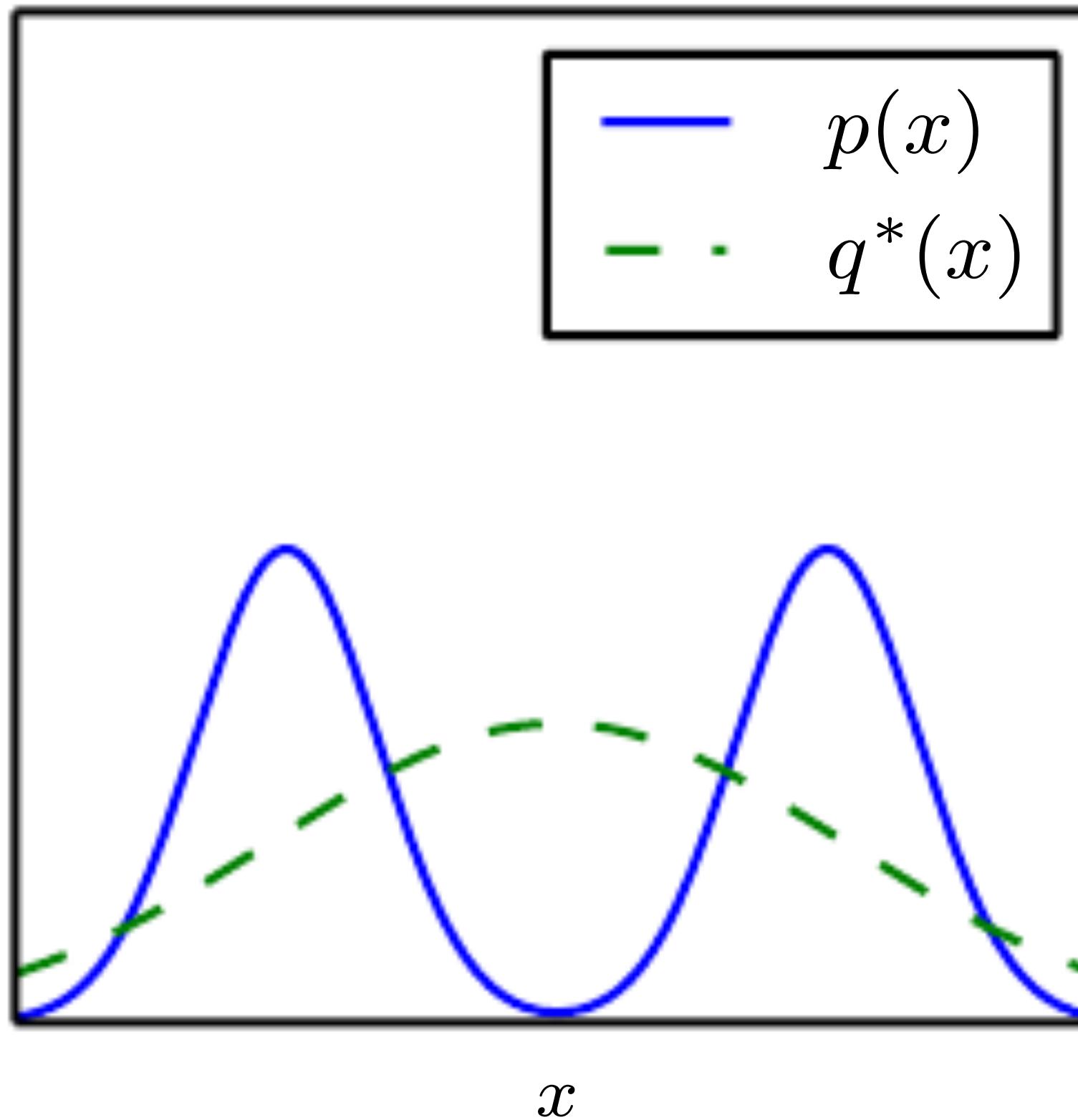
Objective	Model	Data	Task
MD potential energy surface	3N-dim function	DFT energy/force	Generalization
DFT xc energy functional	3-dim functional	QMC/CCSD/...	
Variational free-energy	3N-dim functional	No	Optimization

# Forward KL or Reverse KL ?

## Maximum likelihood estimation

$$q^* = \operatorname{argmin}_q D_{\text{KL}}(p\|q)$$

Probability Density



## Variational free energy

$$q^* = \operatorname{argmin}_q D_{\text{KL}}(q\|p)$$

Probability Density

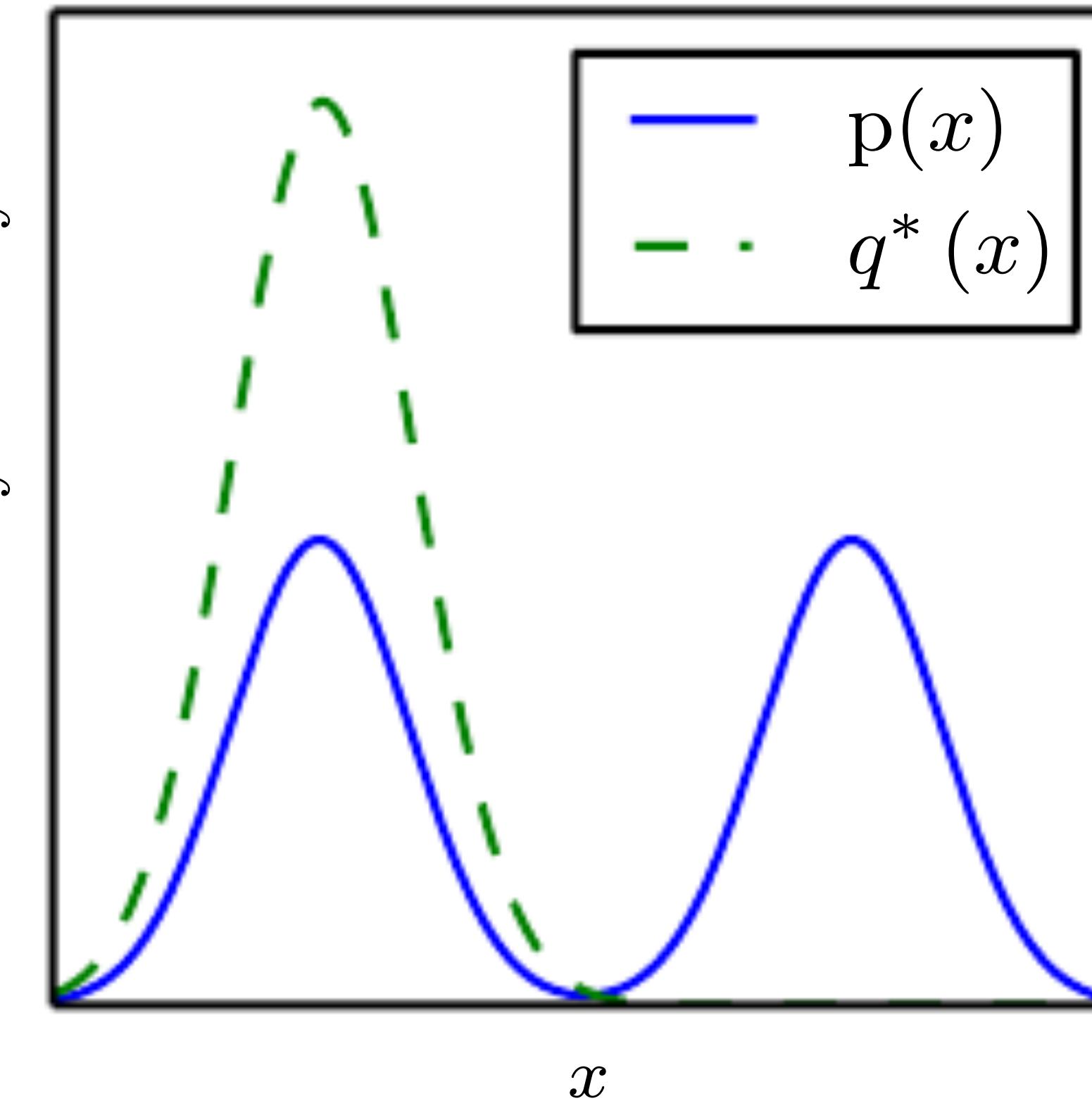
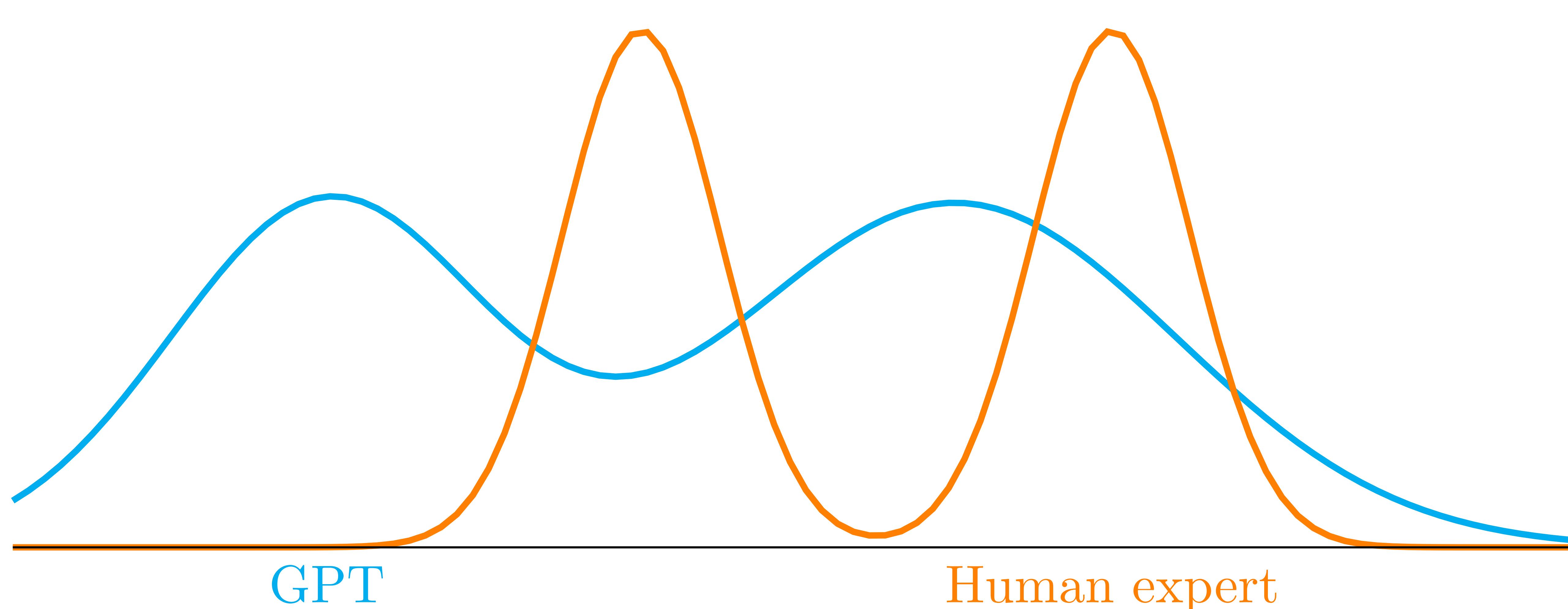


Fig. 3.6, Goodfellow, Bengio, Courville, <http://www.deeplearningbook.org/>



GPT  
“上知天文，下知地理”  
的通才、庸才

填补人类知识的gap v.s. 推进人类知识边界

L

draw two multi-modal distribution using tikz, one distribution is broader, another one is sharper



As an AI language model, I'm unable to create images or drawings directly. However, I can help you create the TikZ code for two multi-modal distributions, one broader and one sharper. You can then compile this code using a LaTeX compiler to generate the illustration.



Here's an example of TikZ code for creating two multi-modal distributions:

latex

 Copy code

```
\documentclass{standalone}
\usepackage{pgfplots}
\usepgfplotslibrary{statistics}

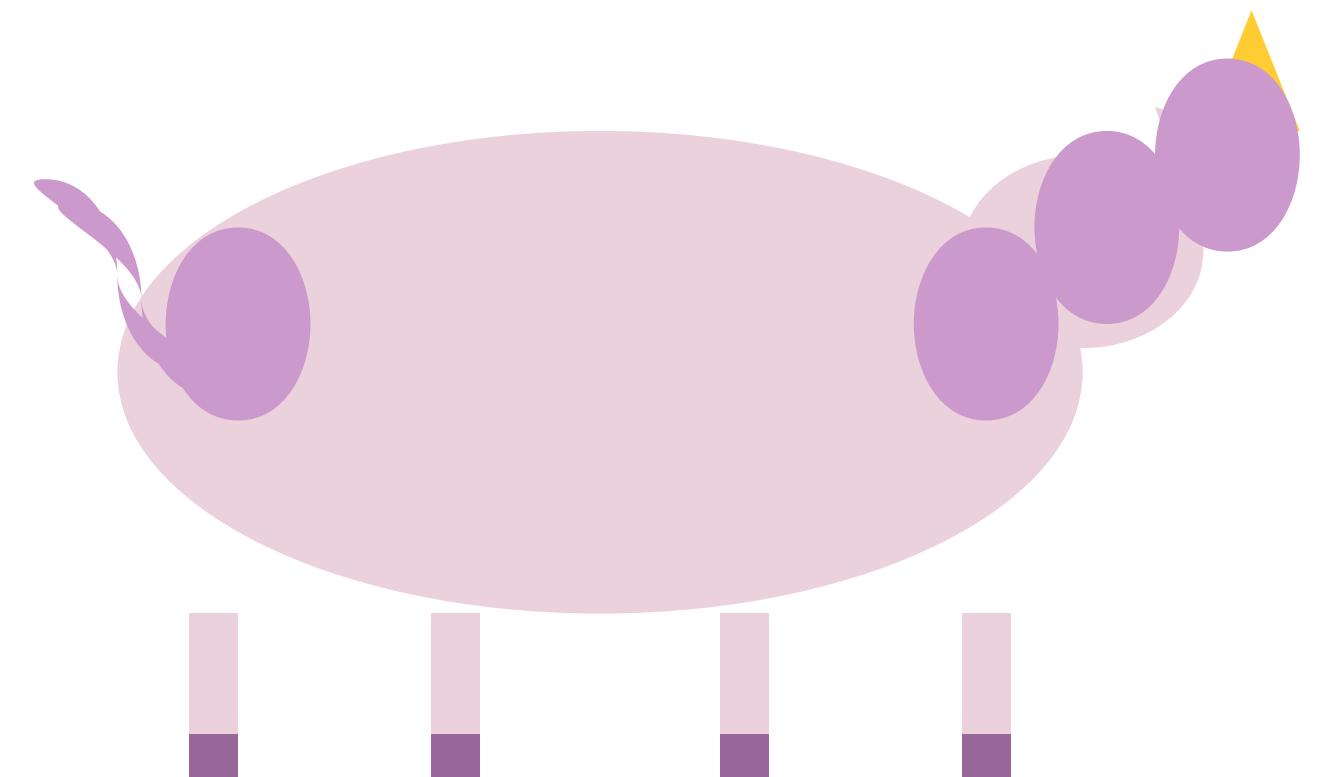
\begin{document}
\begin{tikzpicture}
```

Sparks of Artificial General Intelligence:  
Early experiments with GPT-4

Bubeck et al, 2303.12712

**Prompt:** Draw a unicorn in TiKZ.

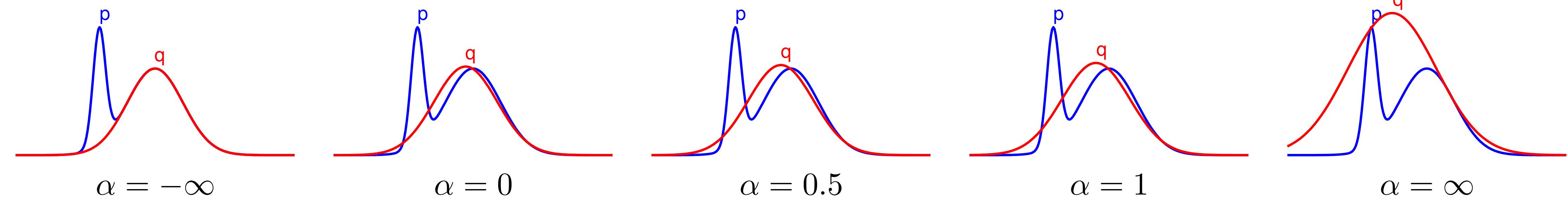
**GPT-4:** [Produces LATEX compiling to following picture.]



# $\alpha$ -divergence

Minka, Microsoft Research Technical Report 2005

$$D_\alpha(p \parallel q) = \frac{\int_x \alpha p(x) + (1 - \alpha)q(x) - p(x)^\alpha q(x)^{1-\alpha} dx}{\alpha(1 - \alpha)}$$



*Fisher divergence*, defined as

$$F(q, p) = \int_{\mathbb{R}^d} \|\nabla \log q(\theta) - \nabla \log p(\theta)\|^2 q(\theta) d\theta,$$

$$D_{-1}(p \parallel q) = \frac{1}{2} \int_x \frac{(q(x) - p(x))^2}{p(x)} dx$$

$$\lim_{\alpha \rightarrow 0} D_\alpha(p \parallel q) = \text{KL}(q \parallel p)$$

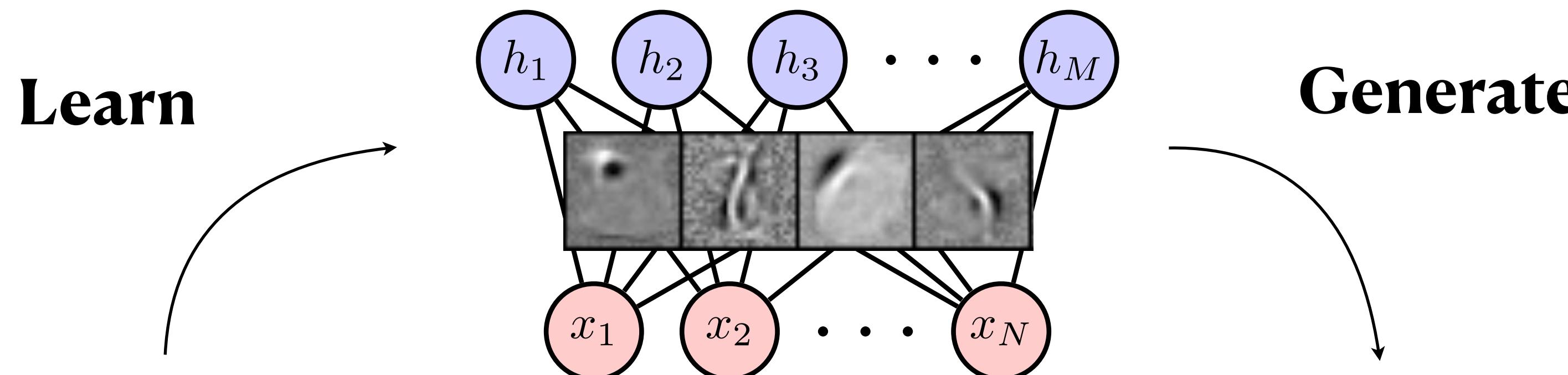
$$D_{\frac{1}{2}}(p \parallel q) = 2 \int_x \left( \sqrt{p(x)} - \sqrt{q(x)} \right)^2 dx$$

$$\lim_{\alpha \rightarrow 1} D_\alpha(p \parallel q) = \text{KL}(p \parallel q)$$

$$D_2(p \parallel q) = \frac{1}{2} \int_x \frac{(p(x) - q(x))^2}{q(x)} dx$$

# Boltzmann machines

$$\mathcal{L} = -\mathbb{E}_{x \sim \text{data}} [\ln p(x)] \quad p(x) = e^{-E(x)}/Z$$



6 2 7 4 2 1 9  
1 2 5 3 0 7 5  
8 1 8 4 2 6 6  
0 7 9 8 6 3 2  
7 5 0 5 7 9 5  
1 8 7 0 6 5 0  
7 5 4 8 4 4 7

$$\nabla \mathcal{L} = \langle \nabla E \rangle_{\text{data}} - \langle \nabla E \rangle_{\text{model}}$$

1 8 3 1 5 7 1  
6 6 3 3 3 1 9  
9 5 8 4 4 1 9  
3 7 7 9 8 7 6  
1 5 3 5 0 2 2  
4 2 5 1 2 4 2  
3 0 5 0 7 0 9

GAUSSIAN-BERNOULLI RBMs WITHOUT TEARS

2210.10318

Renjie Liao<sup>\*1</sup>, Simon Kornblith<sup>2</sup>, Mengye Ren<sup>3</sup>, David J. Fleet<sup>2,4,5</sup>, Geoffrey Hinton<sup>2,4,5</sup>

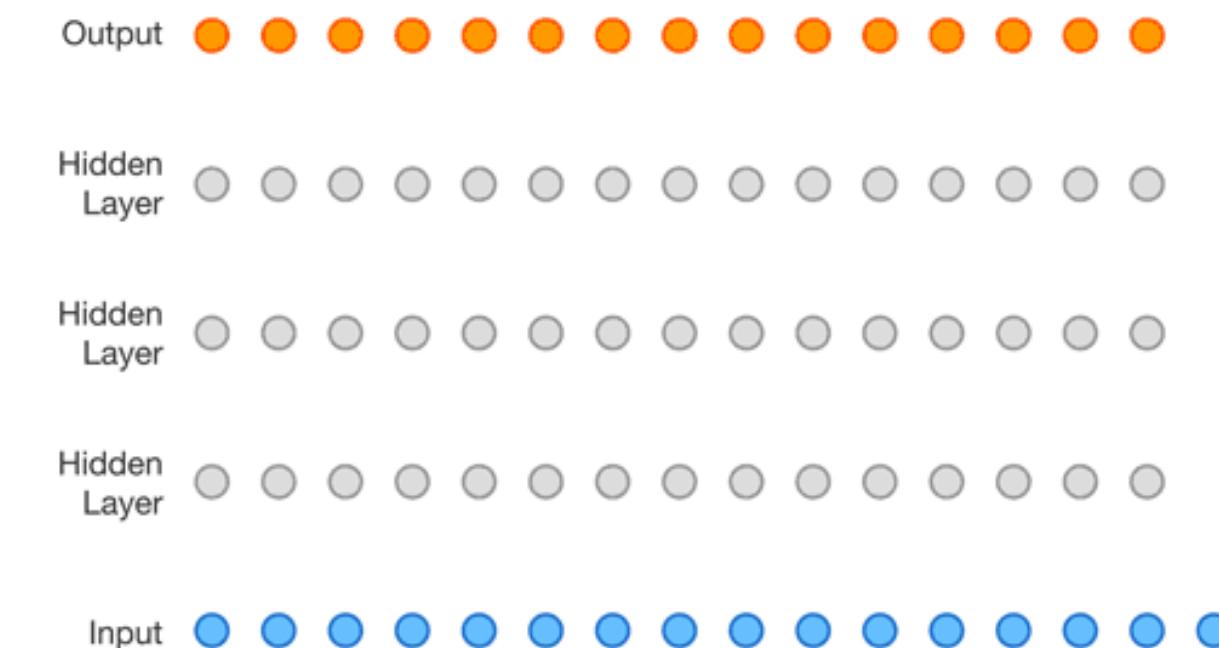
# Autoregressive models

$$p(\mathbf{x}) = p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2)\cdots$$

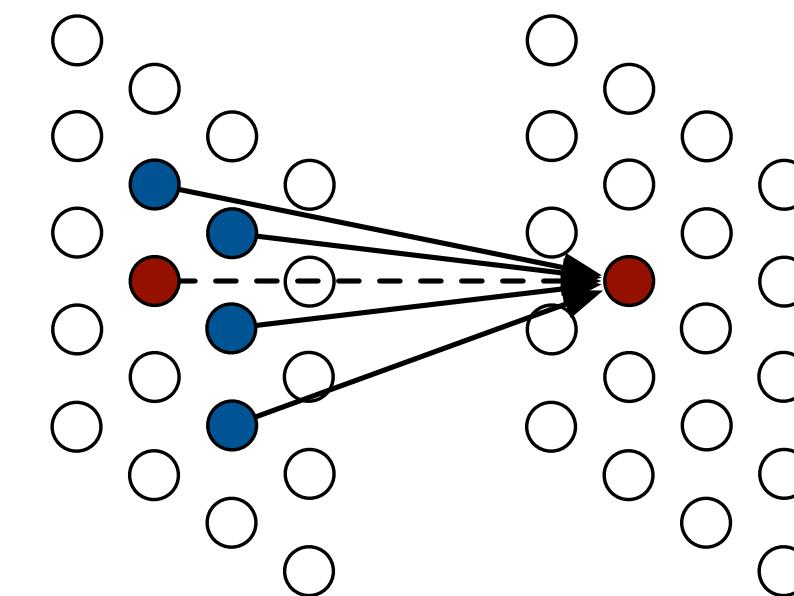
**Language:** GPT 2005.14165



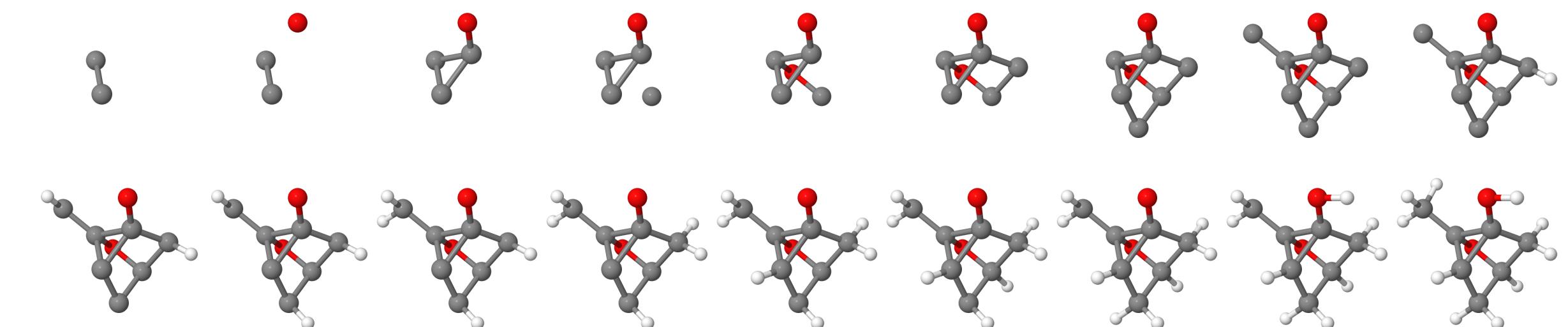
**Speech:** WaveNet 1609.03499



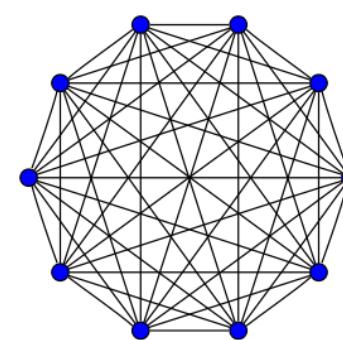
**Image:** PixelCNN 1601.06759



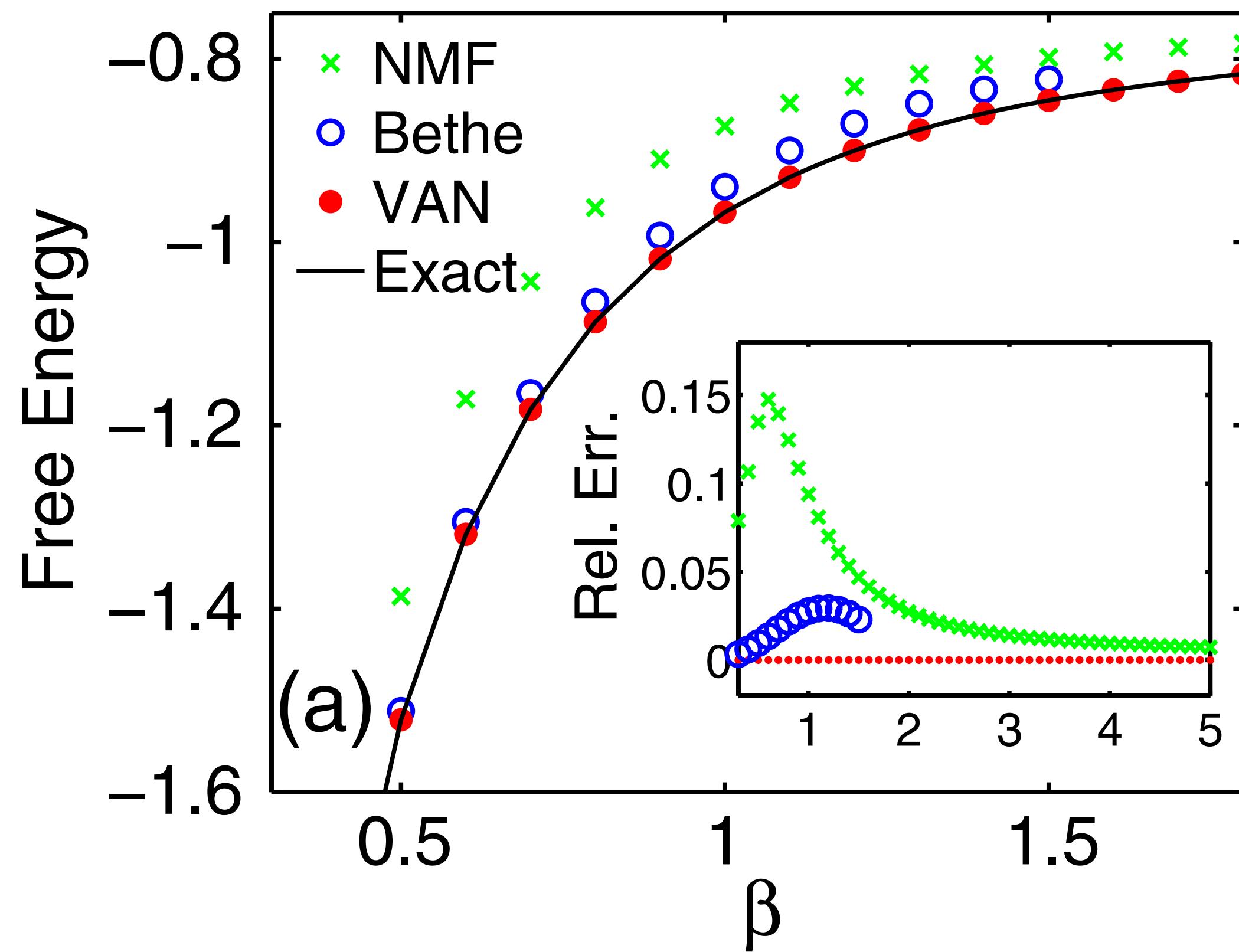
**Molecular graph:** 1810.11347



# Variational autoregressive networks



Sherrington-Kirkpatrick spin glass



Variational autoregressive network

$$p(\mathbf{x}) = \prod_i p(x_i | \mathbf{x}_{<i})$$

[github.com/wdphy16/stat-mech-van](https://github.com/wdphy16/stat-mech-van)

Wu, LW, Zhang, PRL '19

Conventional approaches

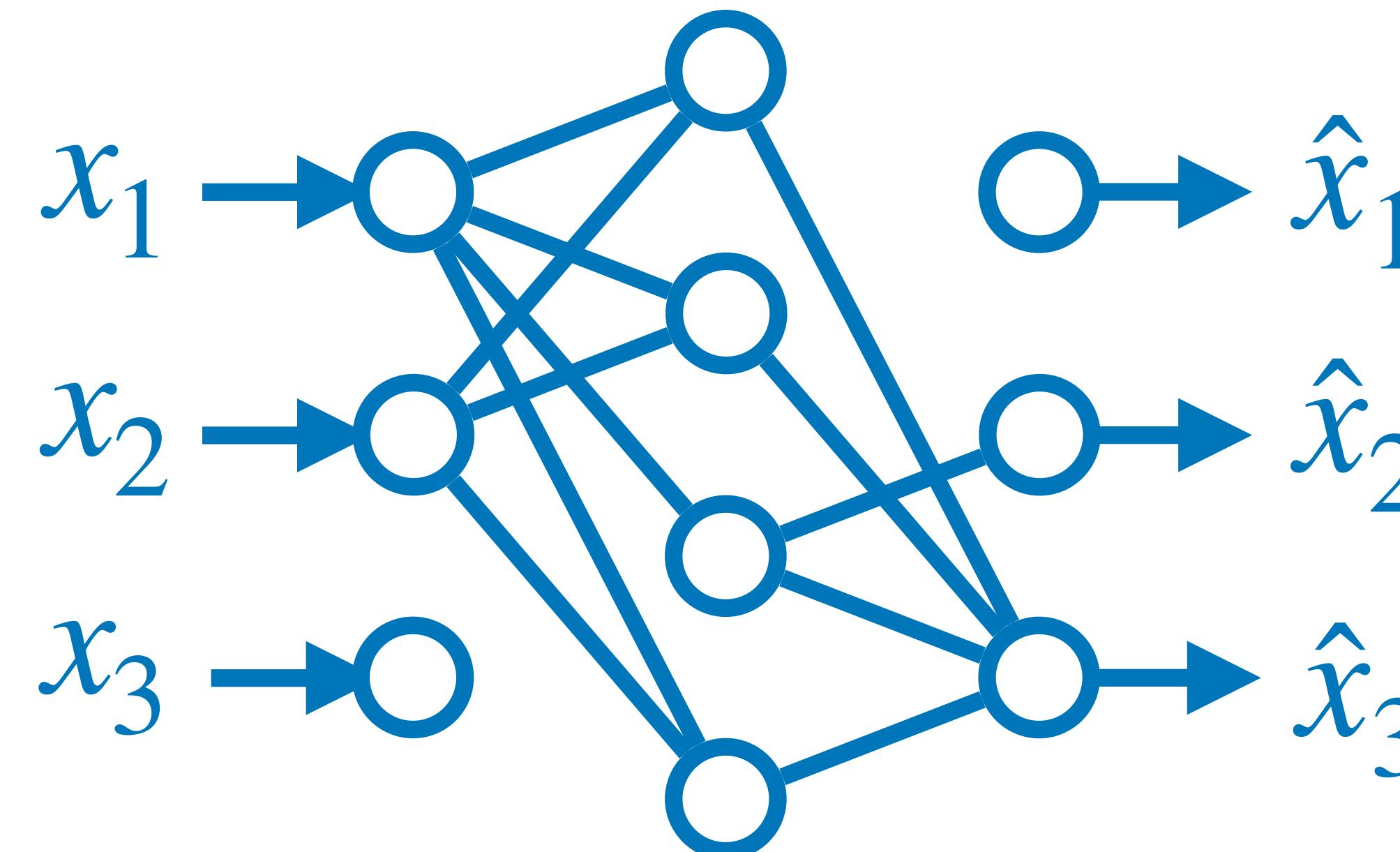
Naive mean-field  
factorized probability

$$p(\mathbf{x}) = \prod_i p(x_i)$$

Bethe approximation  
pairwise interaction

$$p(\mathbf{x}) = \prod_i p(x_i) \prod_{(i,j) \in E} \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

# Implementation: autoregressive masks



Masked Autoencoder  
Germain et al, 1502.03509

$$p(x_1) = \text{Bern}(\hat{x}_1)$$

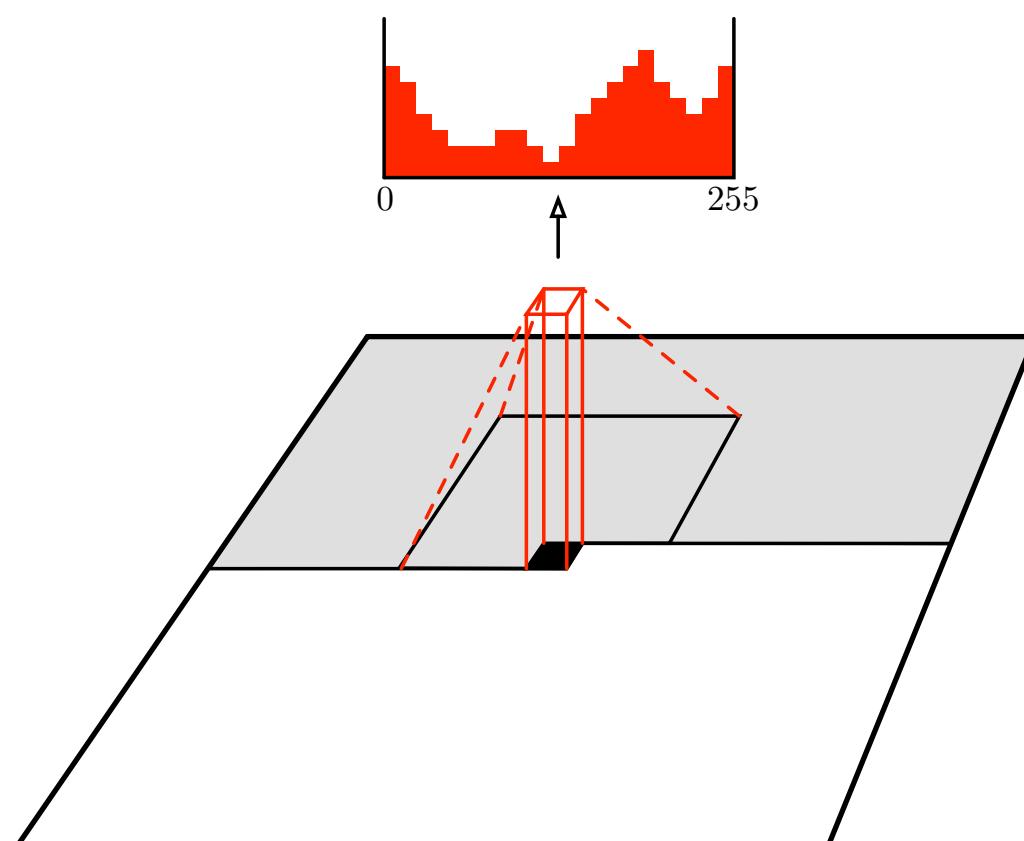
$$p(x_2 | x_1) = \text{Bern}(\hat{x}_2)$$

$$p(x_3 | x_1, x_2) = \text{Bern}(\hat{x}_3)$$

# Implementation: autoregressive masks

Masked CNN kernel

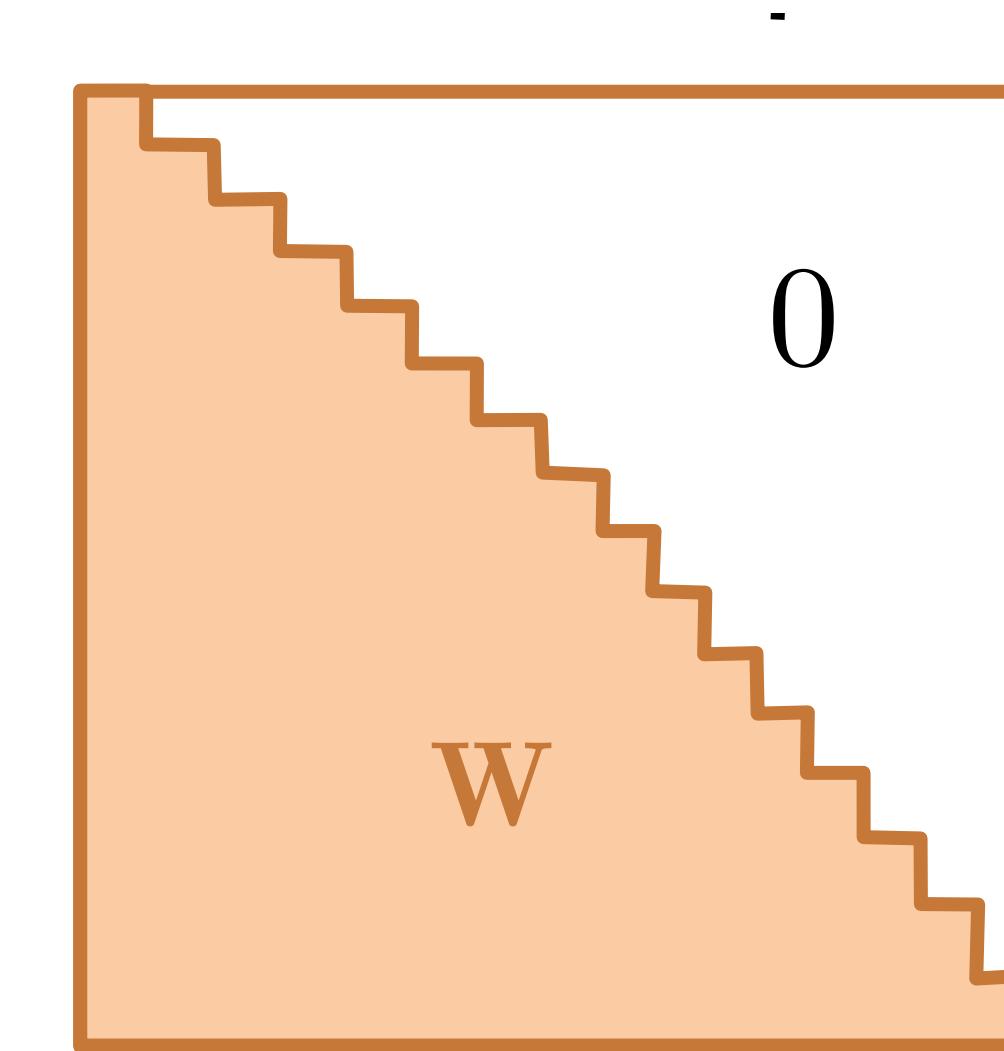
PixelCNN, van den Oord et al, 1601.06759



1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Masked self-attention

Casual transformer, 1706.03762



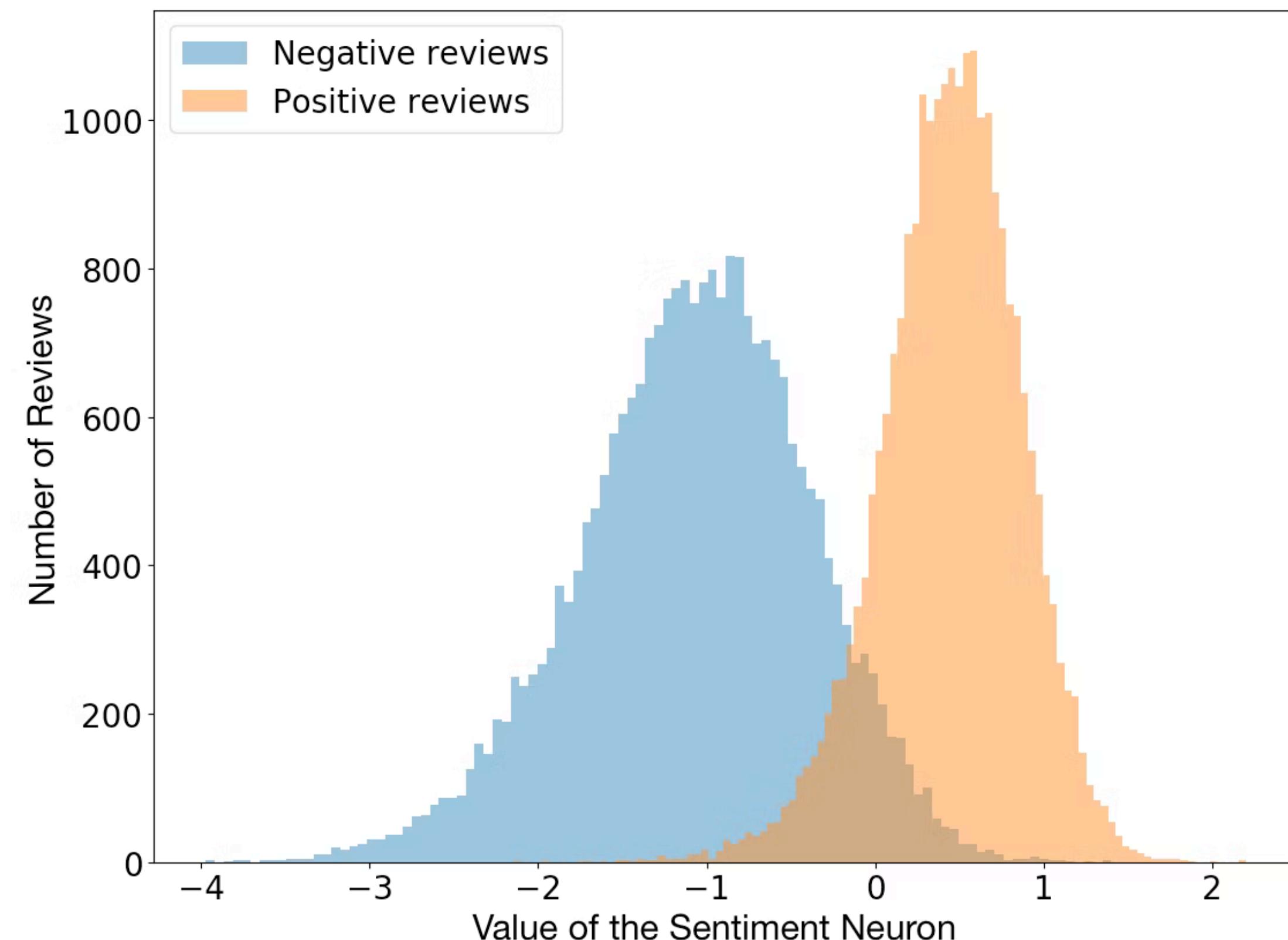
Other ways to implement autoregressive models: recurrent networks

# Learning to Generate Reviews and Discovering Sentiment

Alec Radford<sup>1</sup> Rafal Jozefowicz<sup>1</sup> Ilya Sutskever<sup>1</sup>

We explore the properties of byte-level recurrent language models. When given sufficient amounts of capacity, training data, and compute time, the representations learned by these models include disentangled features corresponding to high-level concepts. Specifically, we find a single unit which performs sentiment analysis. These representations, learned in an unsupervised manner, achieve state of the art on the binary subset of the Stanford Sentiment Treebank. They are also very data efficient. When using only a handful of labeled examples, our approach matches the performance of strong baselines trained on full datasets. We also demonstrate the sentiment unit has a direct influence on the generative process of the model. Simply fixing its value to be positive or negative generates samples with the corresponding positive or negative sentiment.

## “Sentiment neuron”



# Generative Pretraining from Pixels

Mark Chen<sup>1</sup> Alec Radford<sup>1</sup> Rewon Child<sup>1</sup> Jeff Wu<sup>1</sup> Heewoo Jun<sup>1</sup> Prafulla Dhariwal<sup>1</sup> David Luan<sup>1</sup>  
Ilya Sutskever<sup>1</sup>

Inspired by progress in unsupervised representation learning for natural language, we examine whether similar models can learn useful representations for images. We train a sequence Transformer to auto-regressively predict pixels, without incorporating knowledge of the 2D input structure. Despite training on low-resolution ImageNet without labels, we find that a GPT-2 scale model learns strong image representations as measured by linear probing, fine-tuning, and low-data classification. On CIFAR-10, we achieve 96.3% accuracy with a linear probe, outperforming a supervised Wide ResNet, and 99.0% accuracy with full fine-tuning, matching the top supervised pre-trained models. An even larger model trained on a mixture of ImageNet and web images is competitive with self-supervised benchmarks on ImageNet, achieving 72.0% top-1 accuracy on a linear probe of our features.

## “Image GPT”

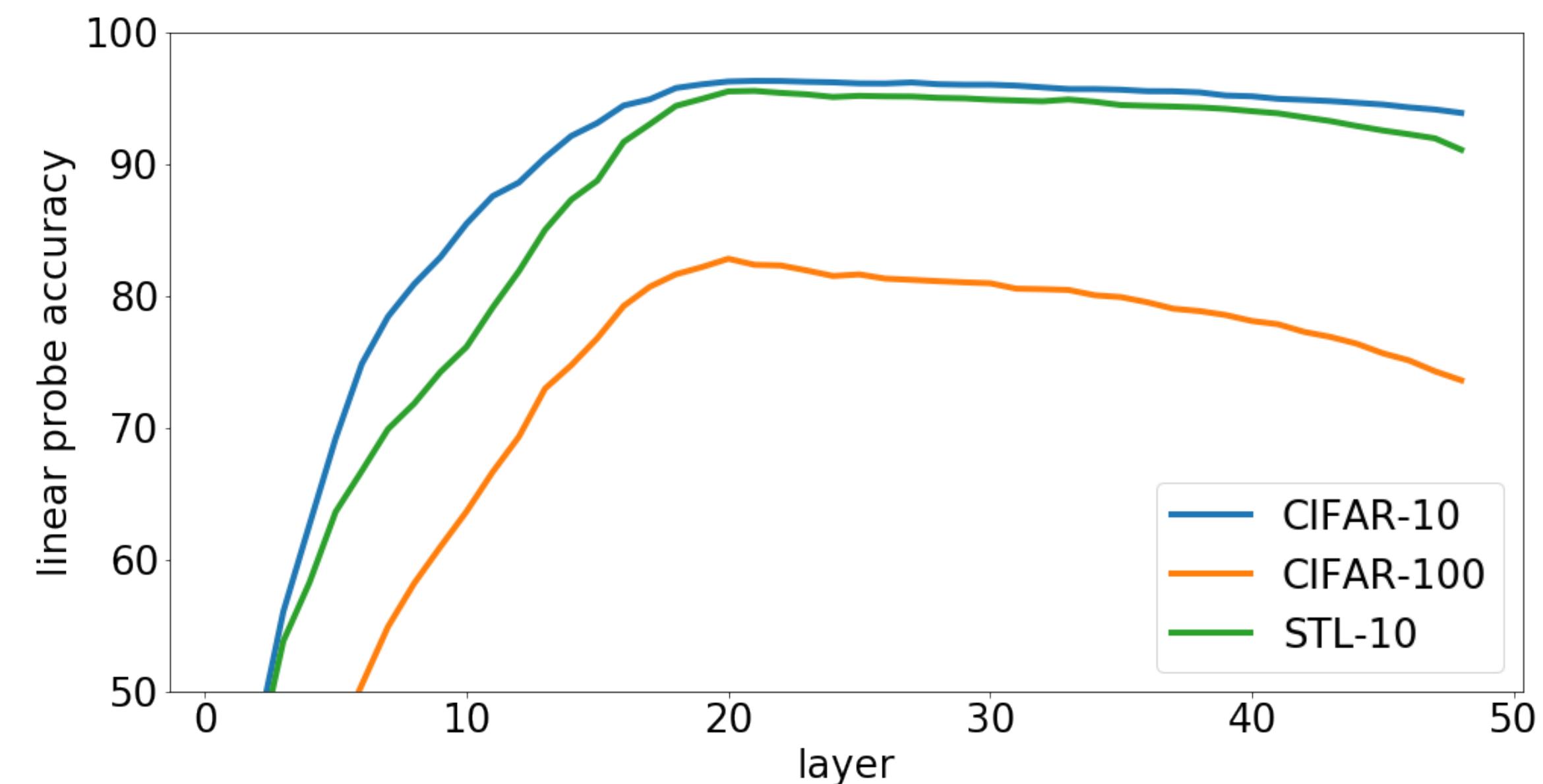
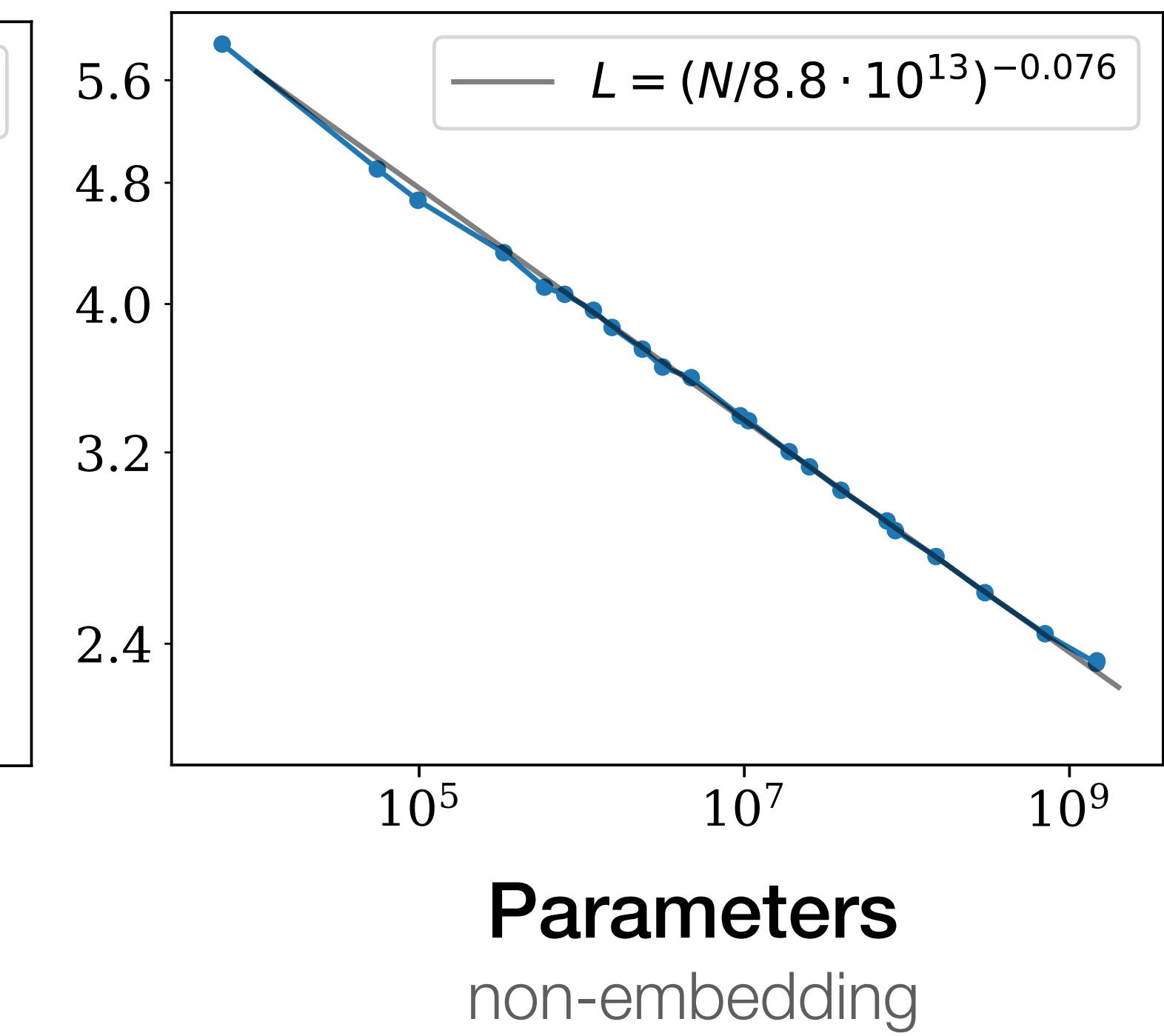
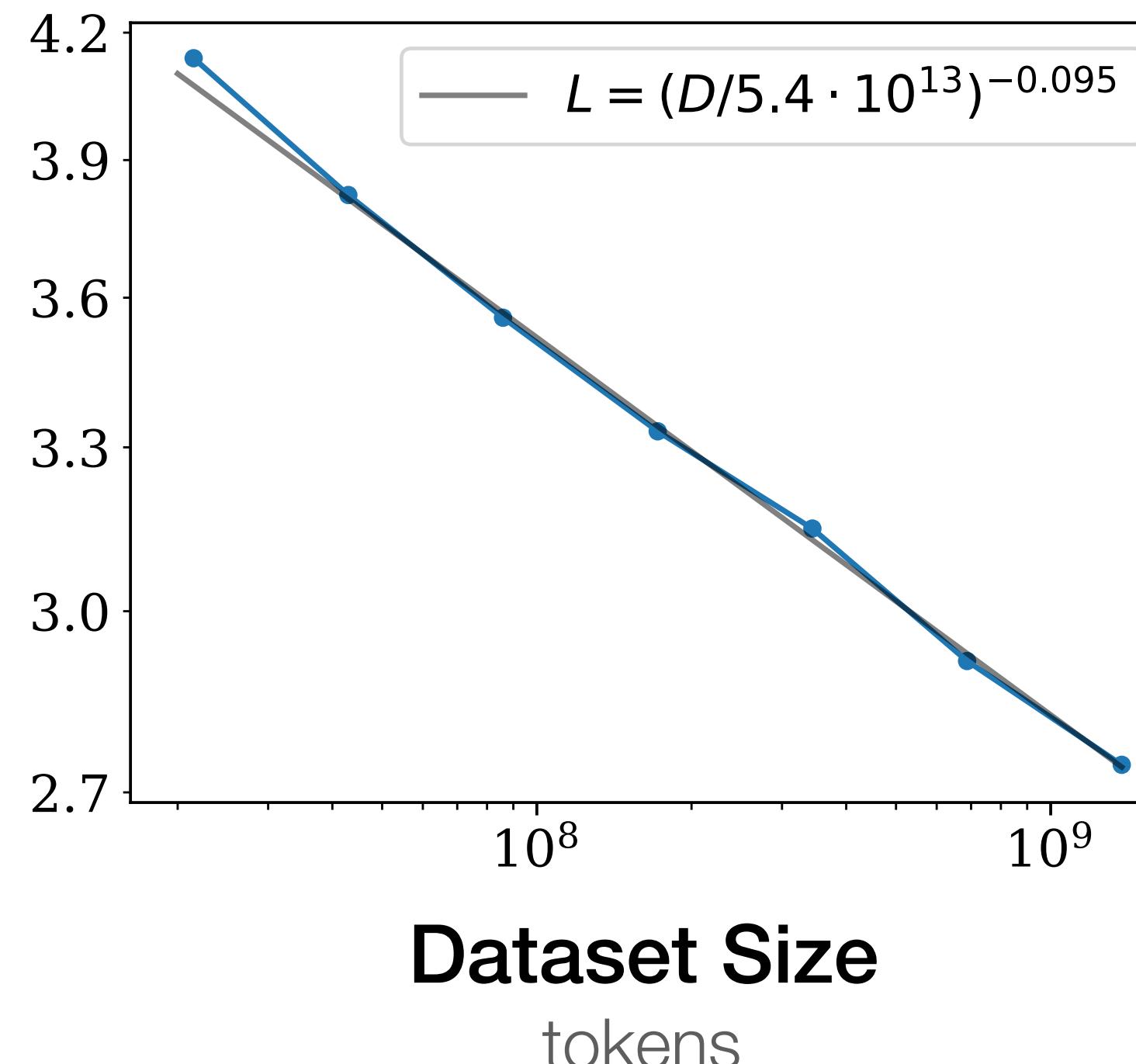
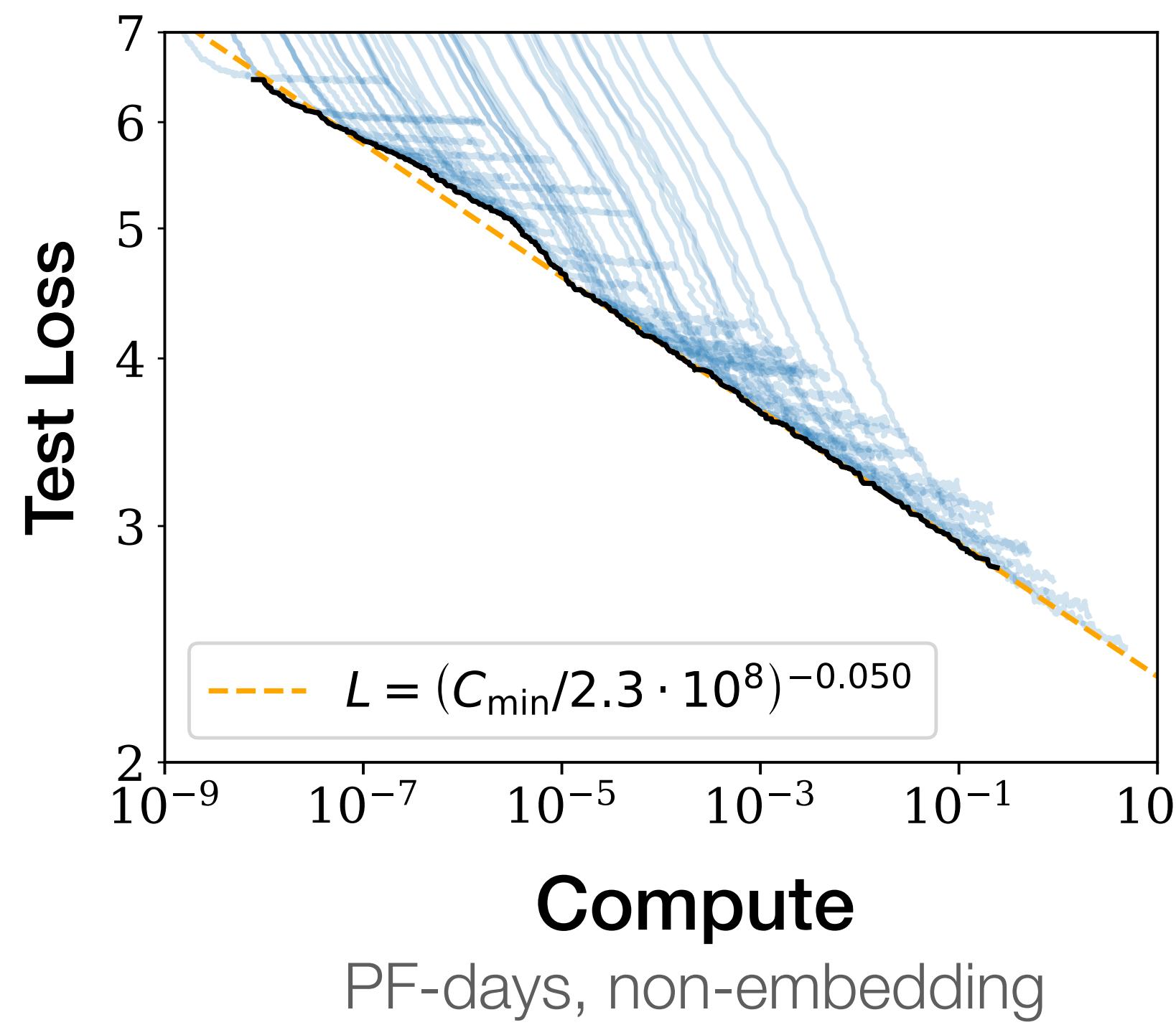


Figure 2. Representation quality depends on the layer from which we extract features. In contrast with supervised models, the best representations for these generative models lie in the middle of the network. We plot this unimodal dependence on depth by showing linear probes for iGPT-L on CIFAR-10, CIFAR-100, and STL-10.

# Scaling law

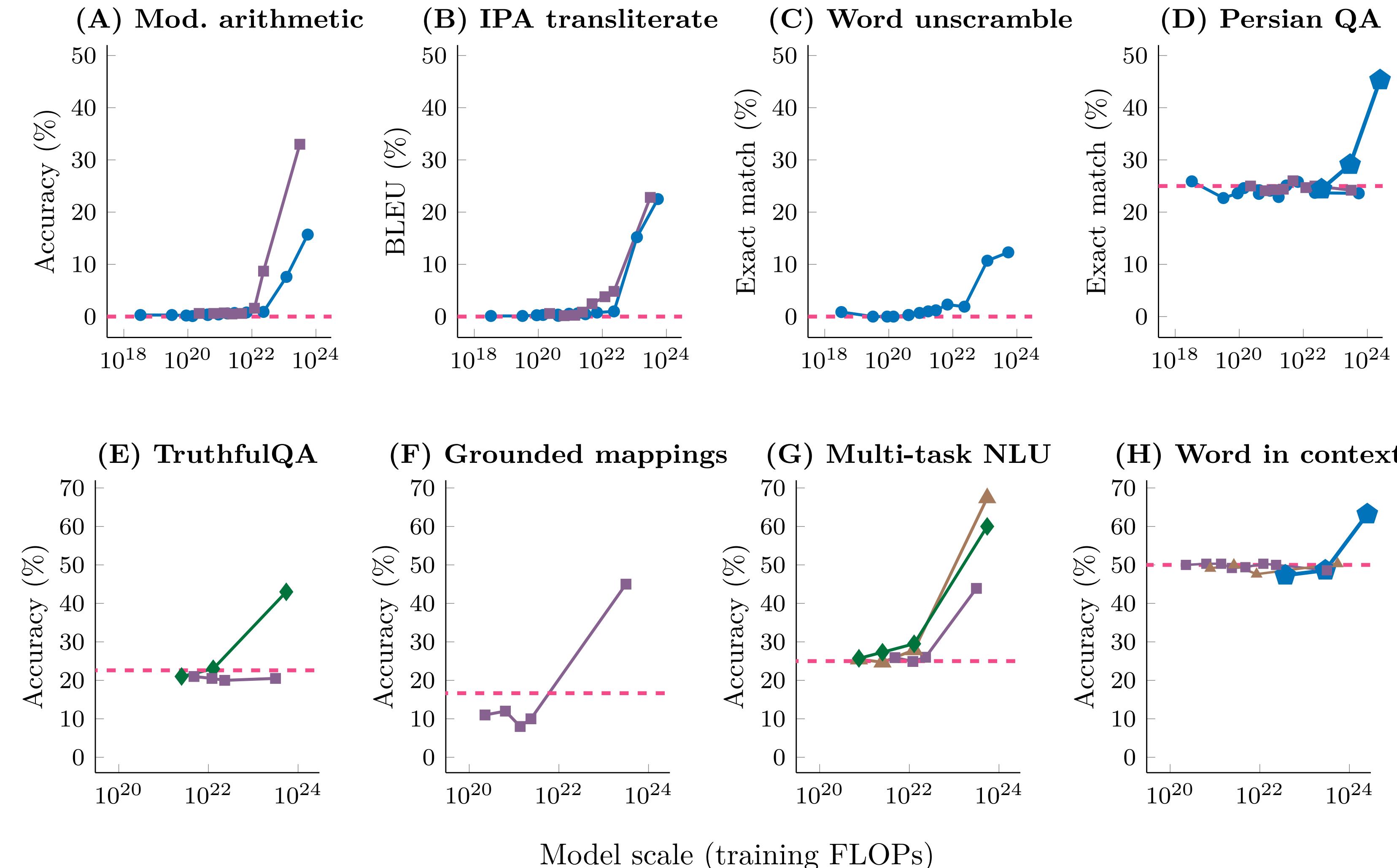
Kaplan 2001.08361



# Emergent abilities: more is different!

Wei et al, 2206.07682  
[https://www.jasonwei.net/  
blog/emergence](https://www.jasonwei.net/blog/emergence)

—●— LaMDA   —■— GPT-3   —◆— Gopher   —▲— Chinchilla   —◆— PaLM   - - - Random



# Variational autoencoders

Kingma, Welling, 1312.6114

Close connection to the variational calculus we have just learned

$$p(\mathbf{x}) = \frac{e^{-\beta E(\mathbf{x})}}{Z}$$

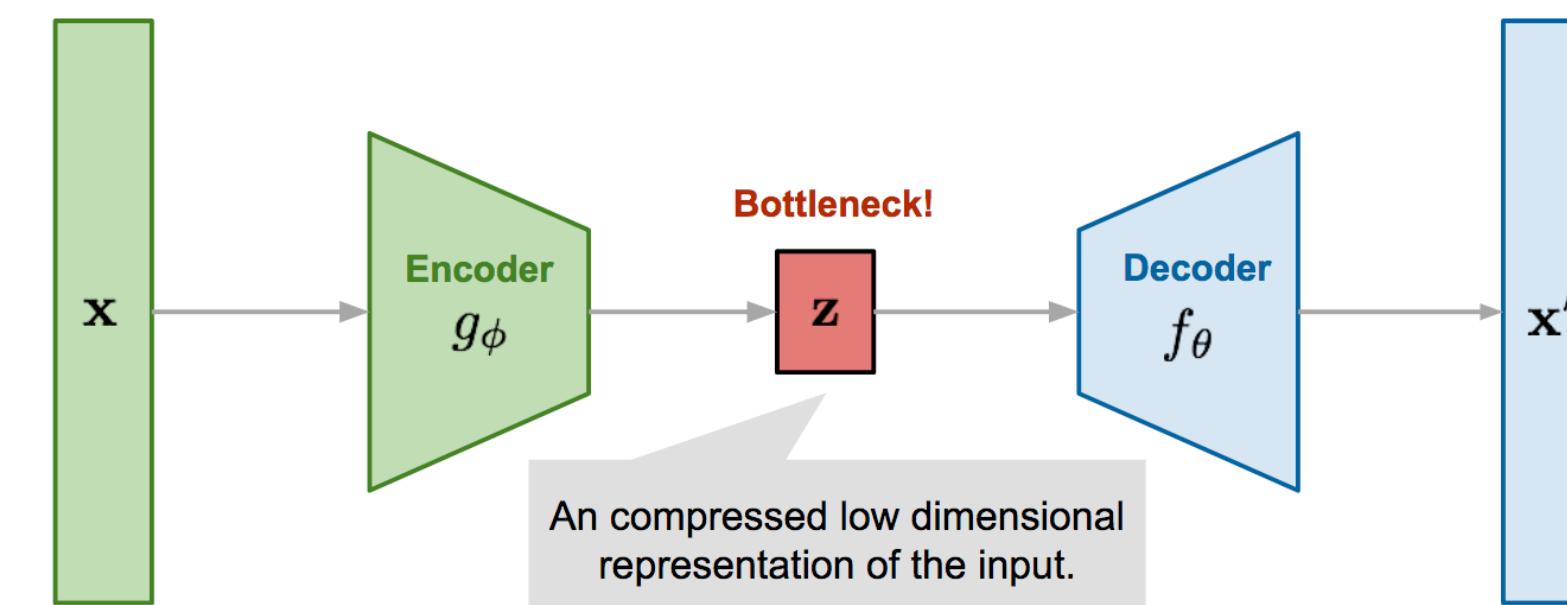
Variational free energy

$$\int d\mathbf{x} q(\mathbf{x}) [\ln q(\mathbf{x}) + \beta E(\mathbf{x})] \geq -\ln Z$$

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

Variational Bayes/Variational inference

$$\int dz q(z | \mathbf{x}) [\ln q(z | \mathbf{x}) - \ln p(\mathbf{x}, z)] \geq -\ln p(\mathbf{x})$$



For each data we introduce

$$\mathcal{L}(\mathbf{x}) = \langle -\ln p(\mathbf{x}, \mathbf{z}) + \ln q(\mathbf{z}|\mathbf{x}) \rangle_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})}, \quad (53)$$

which is a variational upper bound of  $-\ln p(\mathbf{x})$  since  $\mathcal{L}(\mathbf{x}) + \ln p(\mathbf{x}) = \text{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) \geq 0$ . We see that  $q(\mathbf{z}|\mathbf{x})$  provides a variational approximation of the posterior  $p(\mathbf{z}|\mathbf{x})$ . By minimizing  $\mathcal{L}$  one effectively pushes the two distributions together. And the variational free energy becomes exact only when  $q(\mathbf{z}|\mathbf{x})$  matches to  $p(\mathbf{z}|\mathbf{x})$ . In fact,  $-\mathcal{L}$  is called evidence lower bound (ELBO) in variational inference.

We can obtain an alternative form of the variational free energy

$$\mathcal{L}_{\theta, \phi}(\mathbf{x}) = -\langle \ln p_{\theta}(\mathbf{x}|\mathbf{z}) \rangle_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} + \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (54)$$

The first term of Eq. (54) is the reconstruction negative log-likelihood, while the second term is the KL divergence between the approximate posterior distribution and the latent prior. We also be explicit about the network parameters  $\theta, \phi$  of the encoder and decoder.

6 6 6 6 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
8 8 8 8 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0  
9 9 2 2 2 2 2 2 3 5 5 6 0 0 0 0 0 0 0 0 0  
9 9 2 2 2 2 2 2 3 3 5 5 6 0 0 0 0 0 0 0 0  
9 9 4 2 2 2 2 3 3 3 5 5 5 8 8 8 8 8 8 8 8  
9 9 9 2 2 2 2 3 3 3 3 5 5 5 5 5 5 5 5 5 5  
9 9 9 9 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5 5 5  
9 9 9 9 9 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5  
9 9 9 9 9 9 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5  
9 9 9 9 9 9 3 3 3 3 3 3 3 3 3 8 8 8 8 8 8  
9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 8 8  
9 8

Learned MNIST  
manifold

# Chemical design using continuous latent variables

Gomez-Bombarelli et al, 1610.02415

