Generative models for physicists

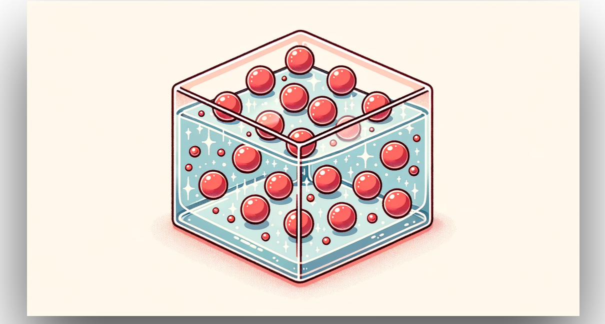
Lei Wang (王磊)
Institute of Physics, CAS
https://wangleiphy.github.io

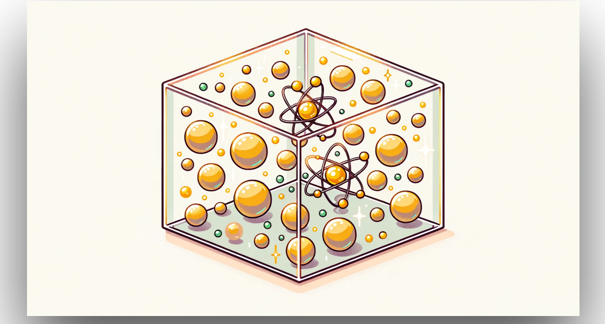
Plan

1. Motivations

2. Generative models and their physics genes

3. Applications: electron gases and dense hydrogen





Science is more than fitting, so is machine learning

Discriminative learning

$$y = f(\boldsymbol{x})$$

or $p(y \mid \boldsymbol{x})$

Generative learning

$$p(\boldsymbol{x}, y)$$

Science is more than fitting, so is machine learning

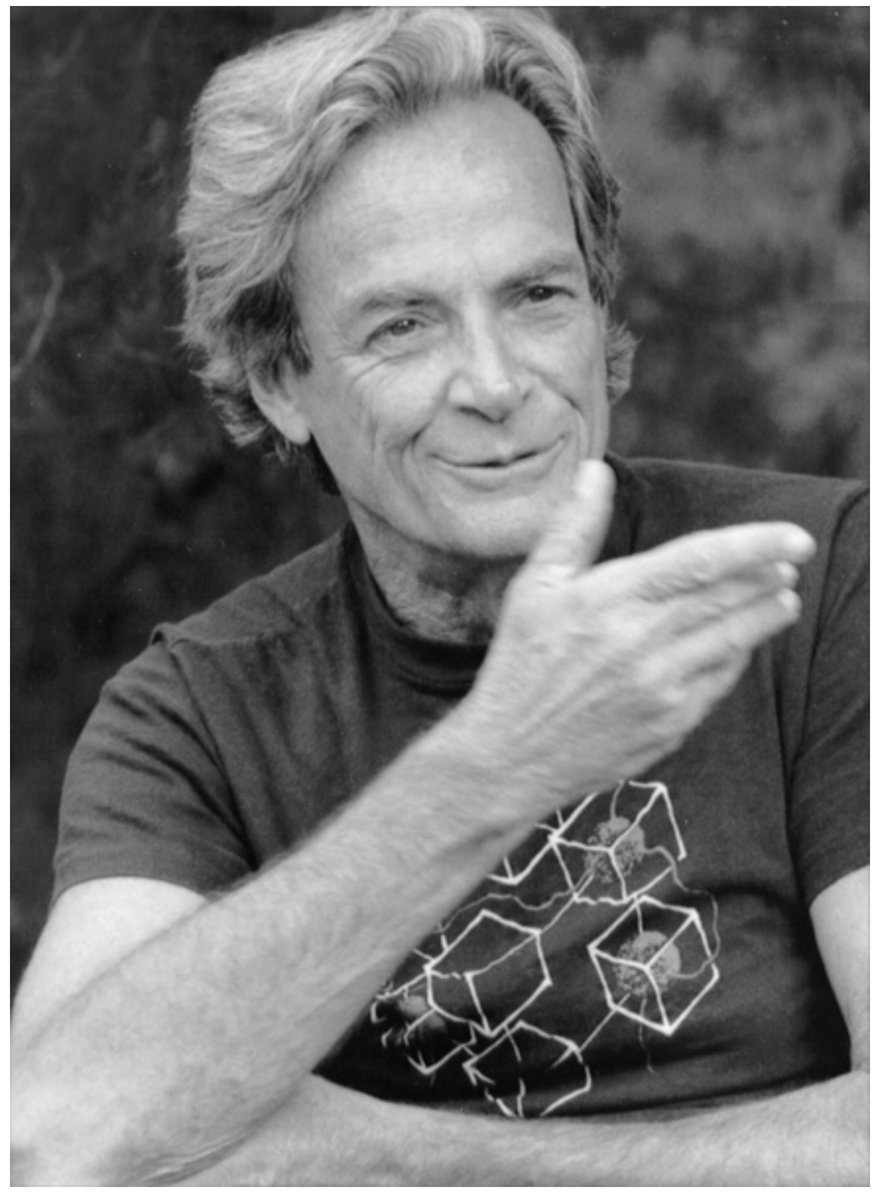
Discriminative learning

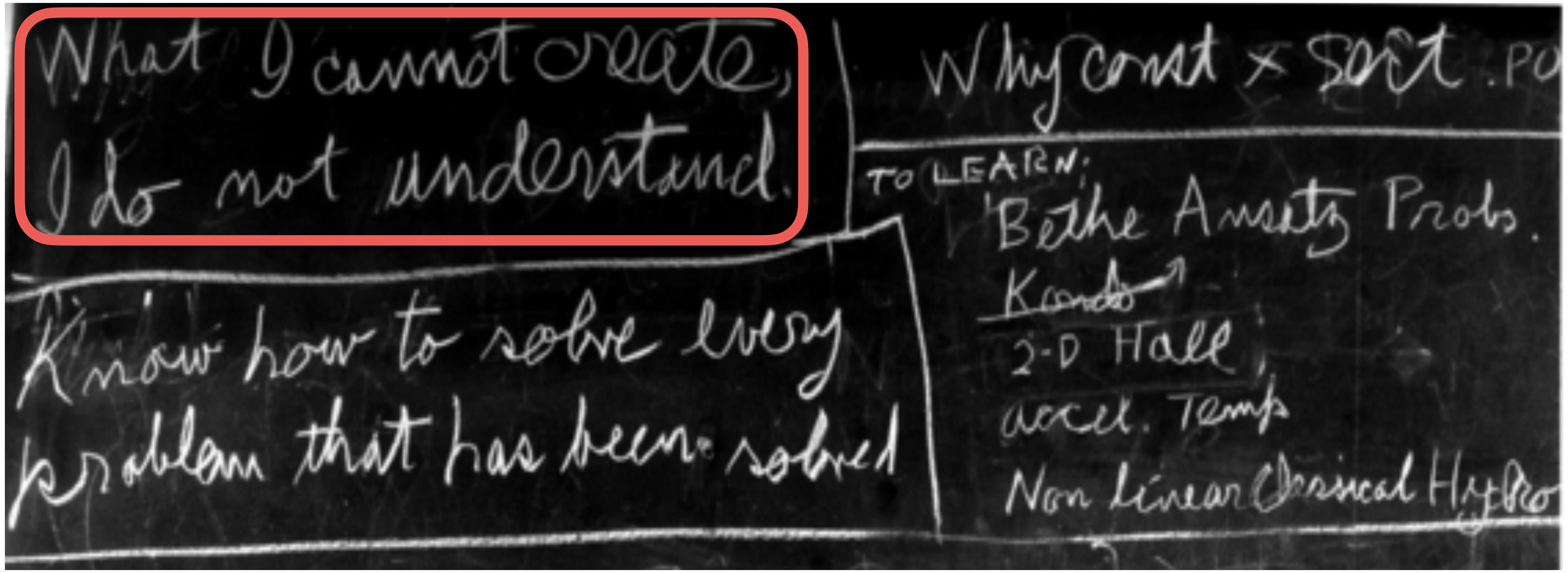
$$y = f(\boldsymbol{x})$$

or $p(y\,|\,\boldsymbol{x})$

Generative learning

$$p(\boldsymbol{x}, y)$$





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

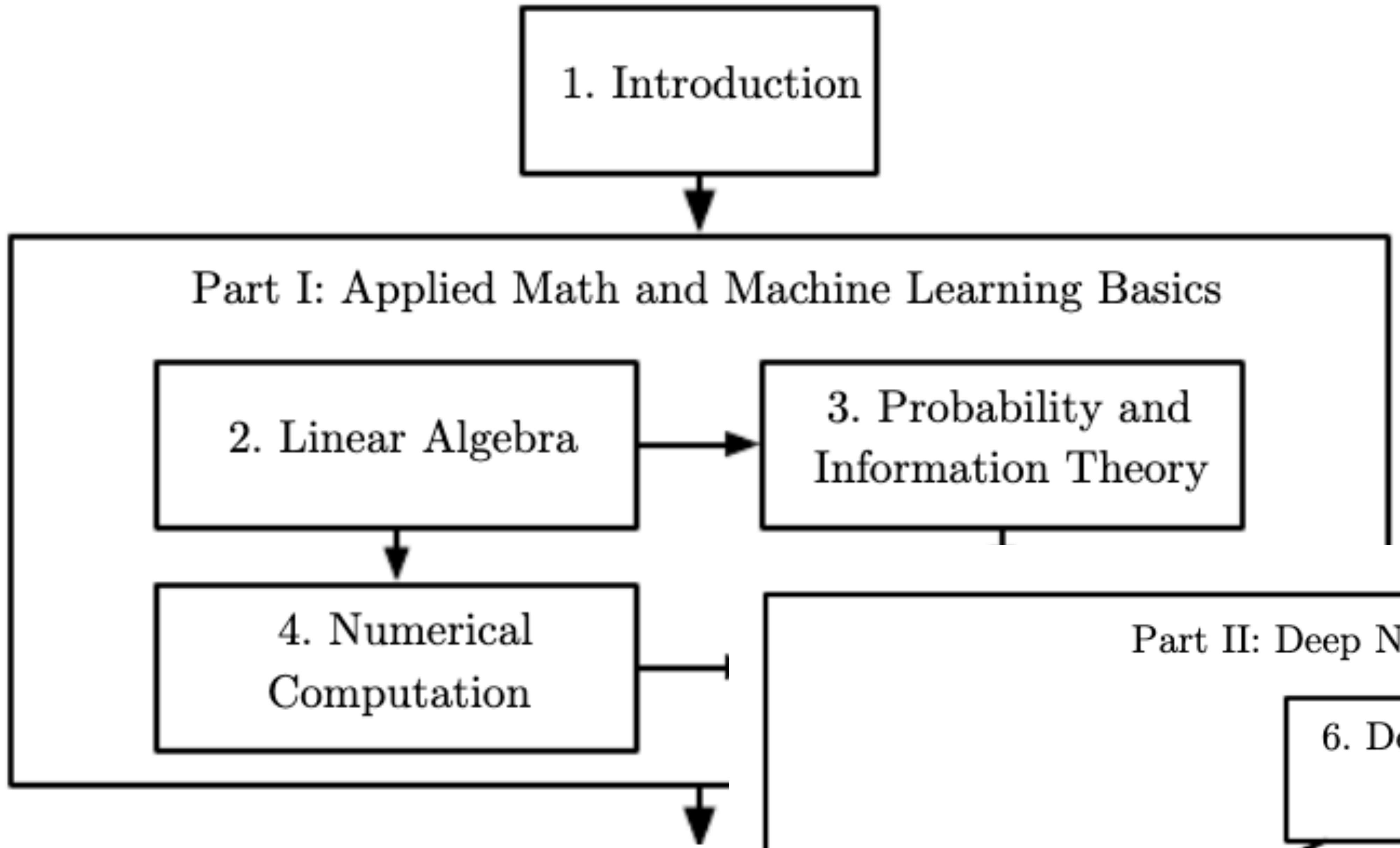
Welcome, Pieter and Shivon!
April 26, 2016 — Announcements

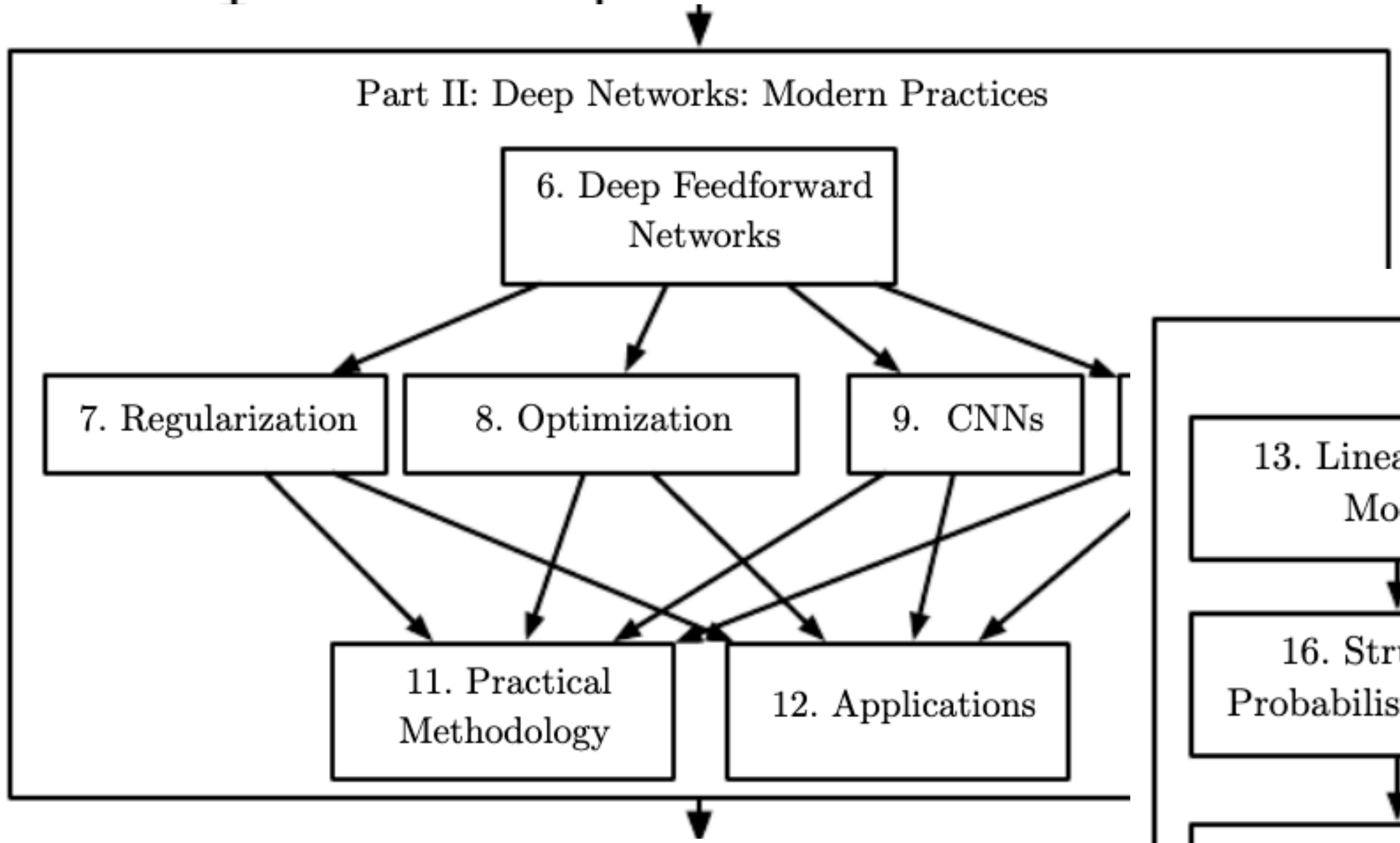
Team++
March 31, 2016 — Announcements

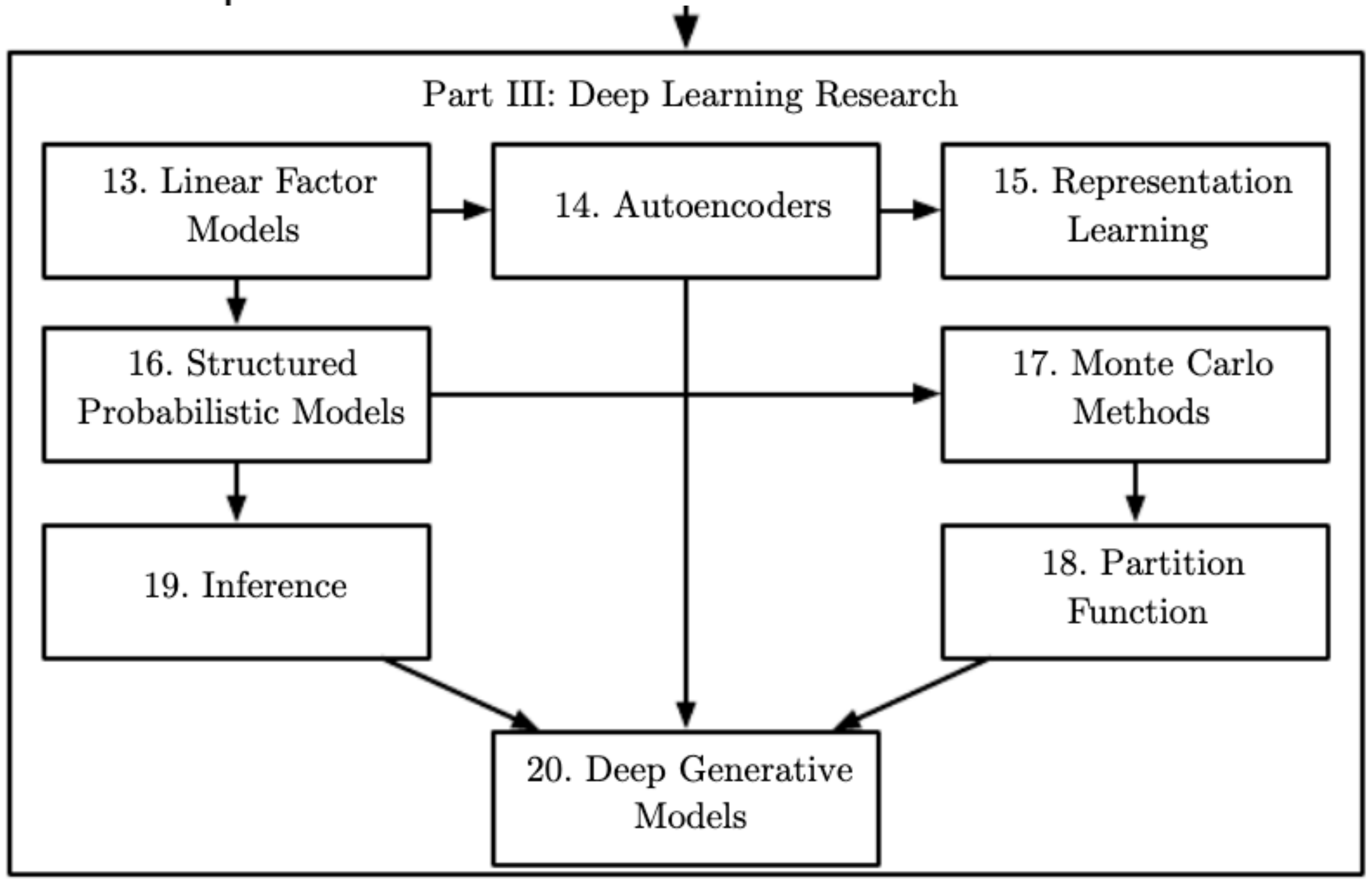
Introducing OpenAI
December 11, 2015 — Announcements

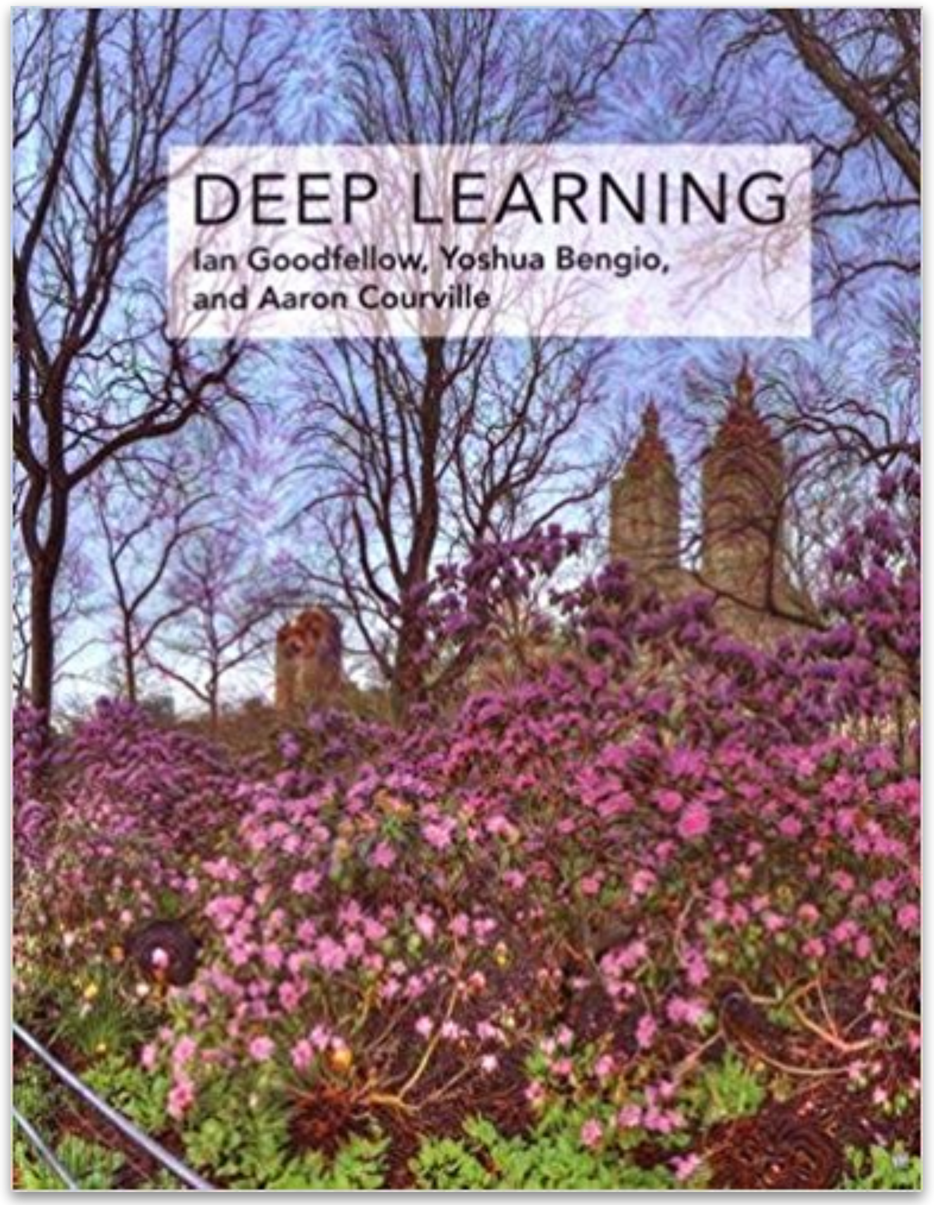
https://openai.com/blog/

“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.”

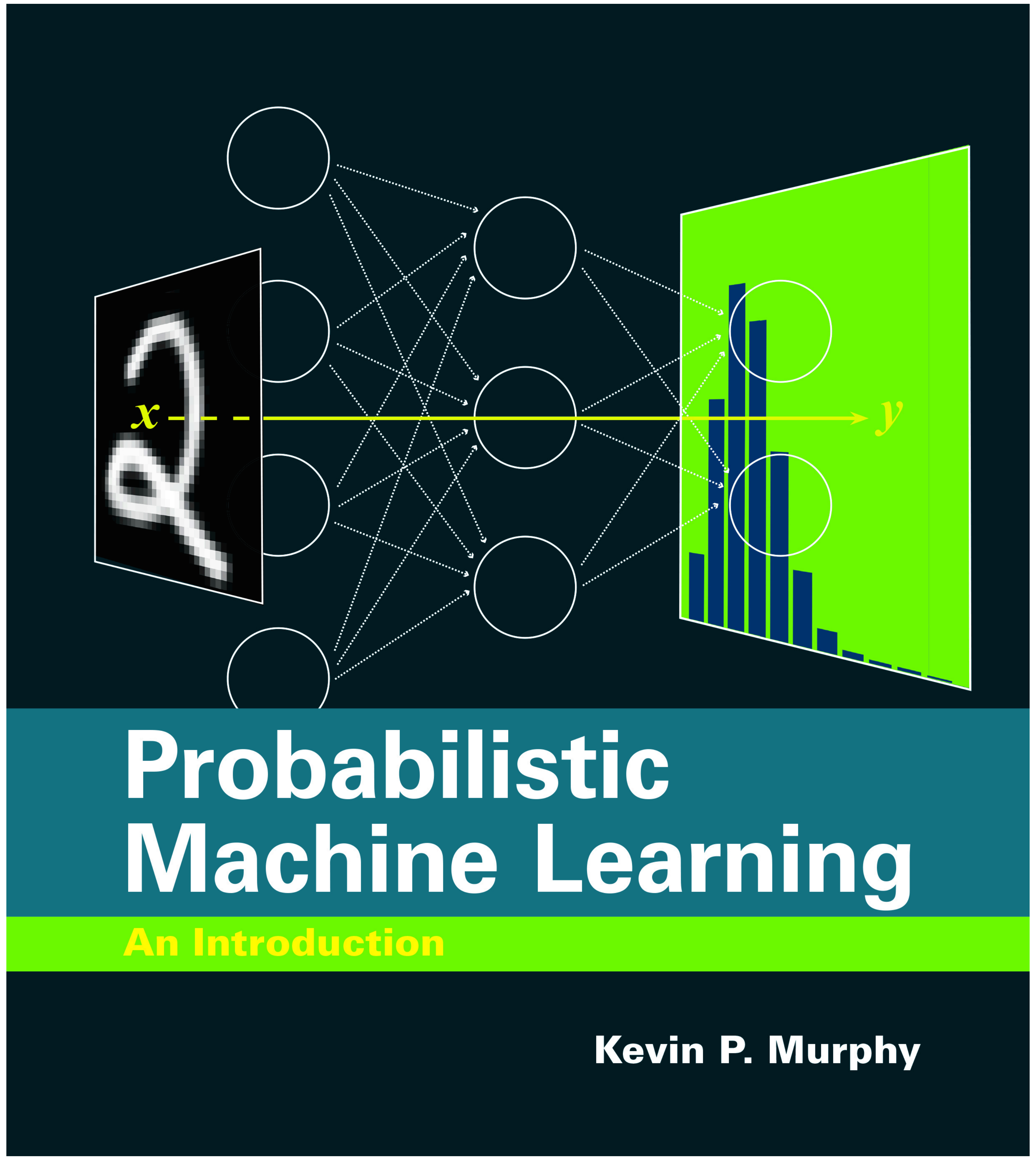


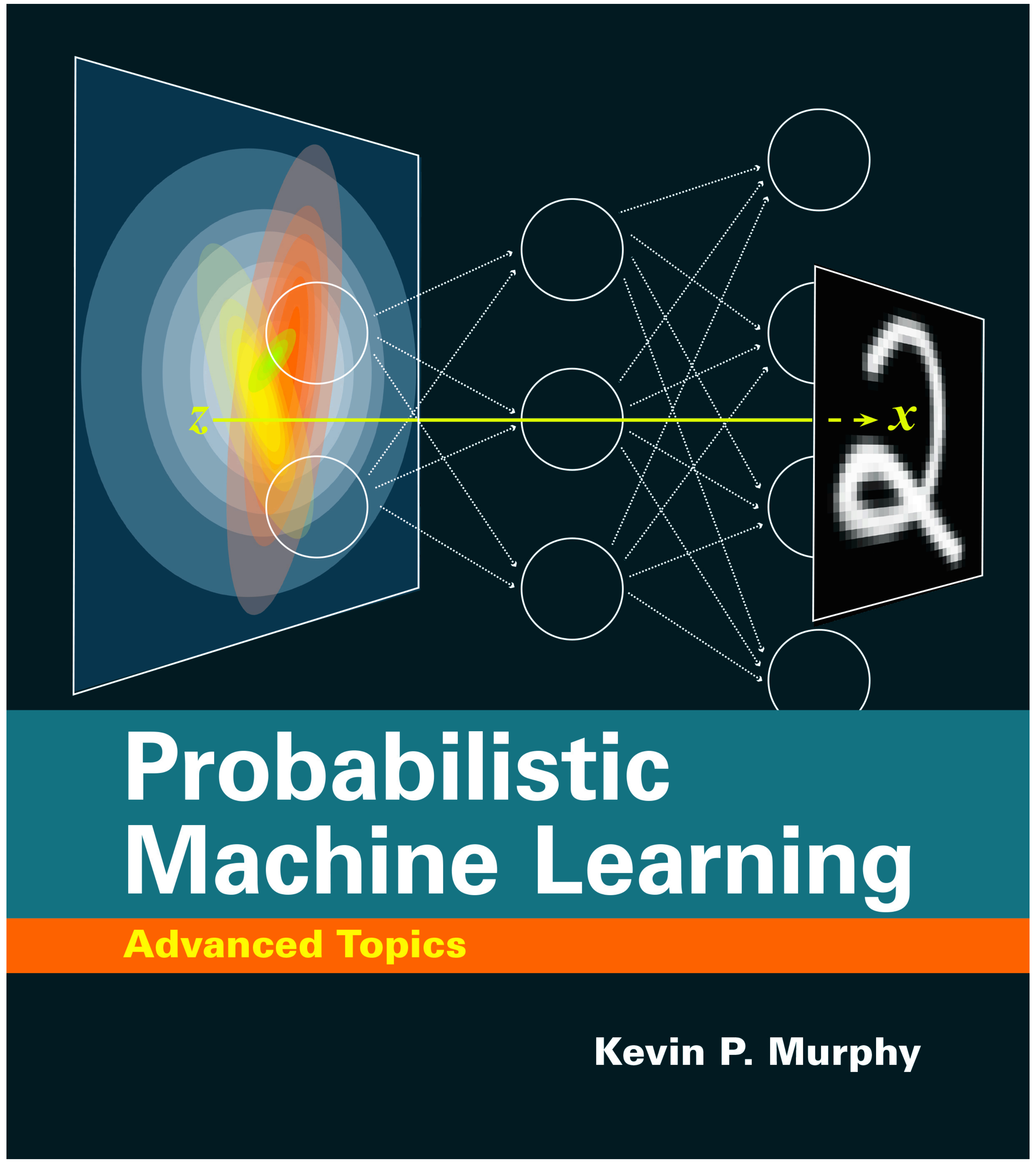






"Part III is the most important for a researcher—someone who wants to understand the breadth of perspectives that have been brought to the field of deep learning, and push the field forward towards true artificial intelligence."

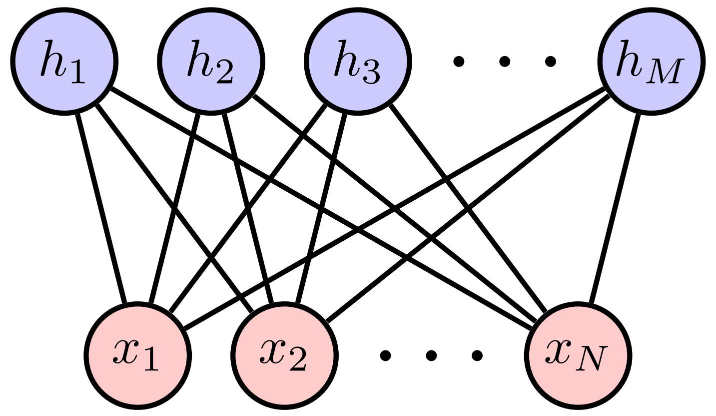


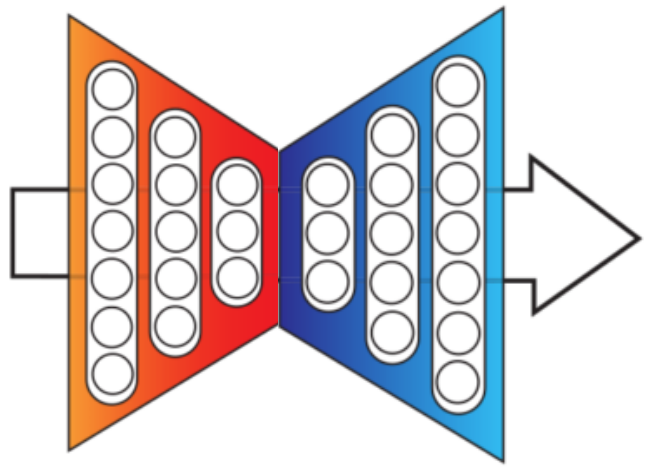


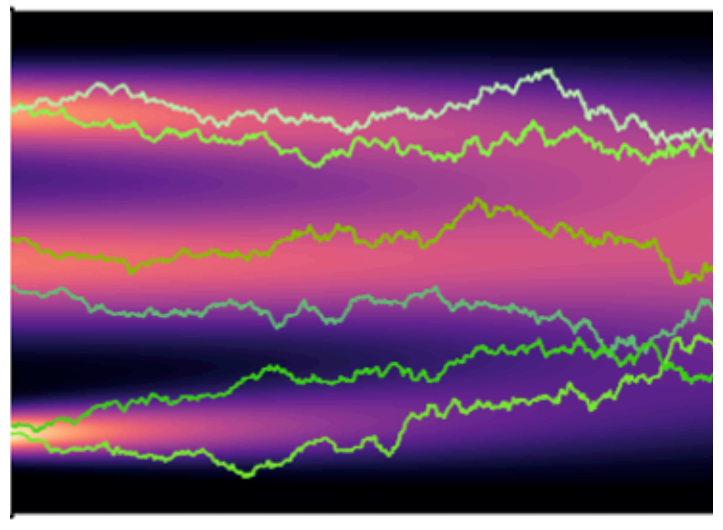
2022 (855 pages)

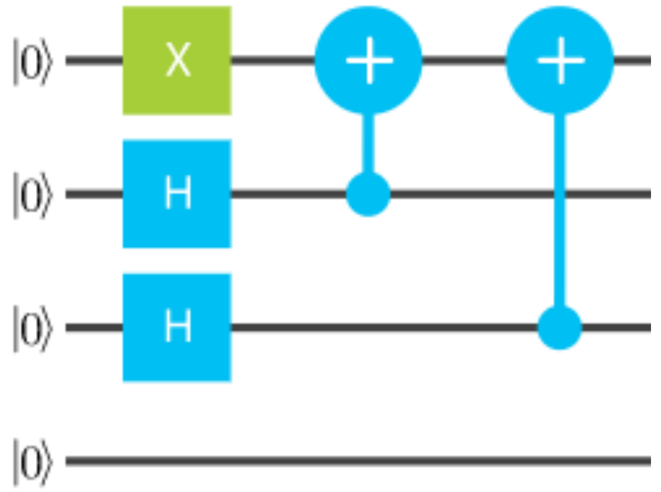
2023 (1352 +332 pages)

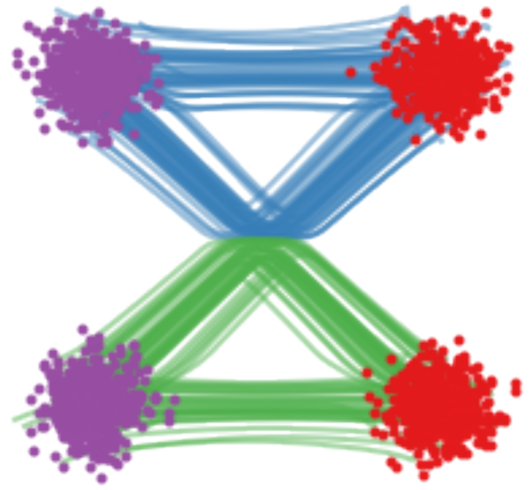
https://probml.github.io/pml-book/











Boltzmann Machine
1985

Variational Autoencoder
2013

Diffusion Model
2015

Born Machine
2017

Flow Matching
2022

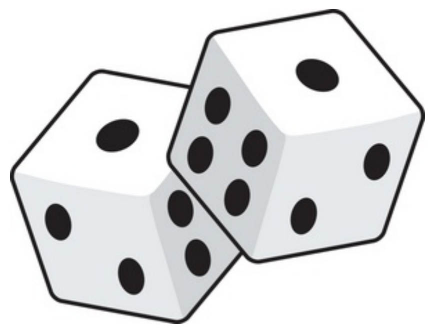
Monte Carlo
Ising model

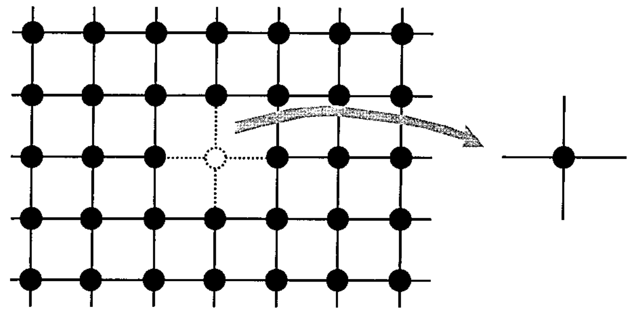
Variational
mean field

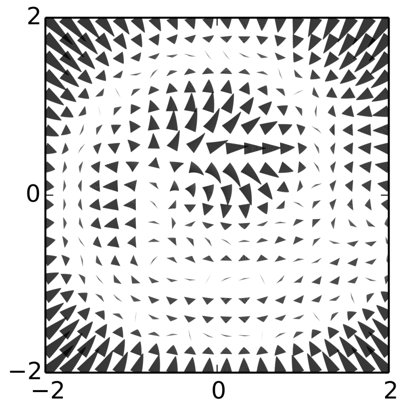
Nonequilibrium
thermodynamics

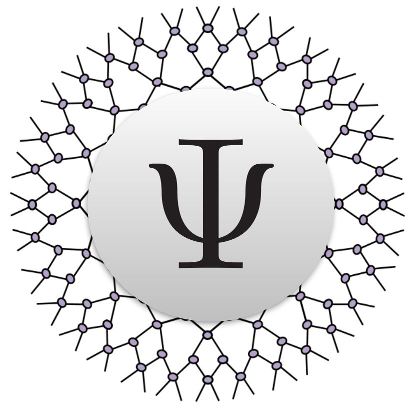
Tensor networks
Quantum circuits

Fluid optimal
transportation









$$\frac{\partial p(\boldsymbol{x},t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x},t)\boldsymbol{v}\right] = 0$$

1. **Leverage the power of modern generative models for physics**
2. **Statistical, quantum, fluid, ... physics insights into generative models**

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 **P**retrained **T**ransformer

$$\text{text} \sim p(\text{text} \mid \text{prompt})$$

Galactica, ChemGPT, MaterBERT, ChemCrow, MatChat...

You may ask (prompts):

"Tell me why this hypothesis is wrong"

"Tell me why my treatment idea won't work"

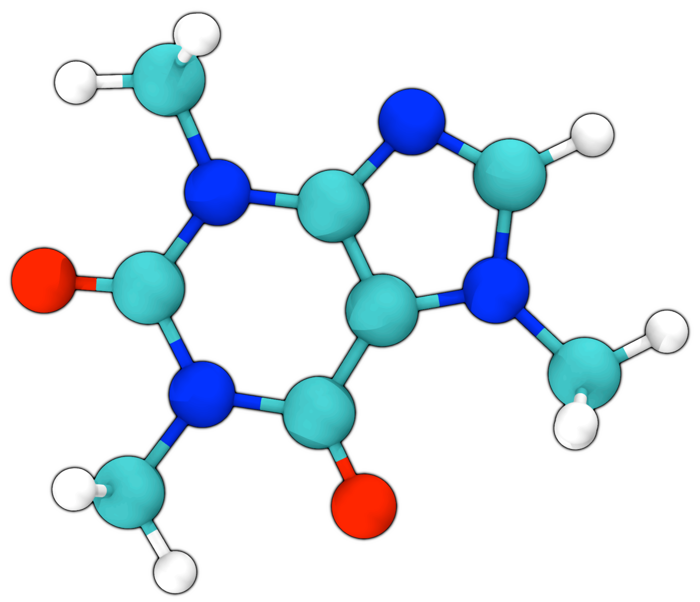
"Generate a new treatment idea"

"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"

Language = anything you can tokenize

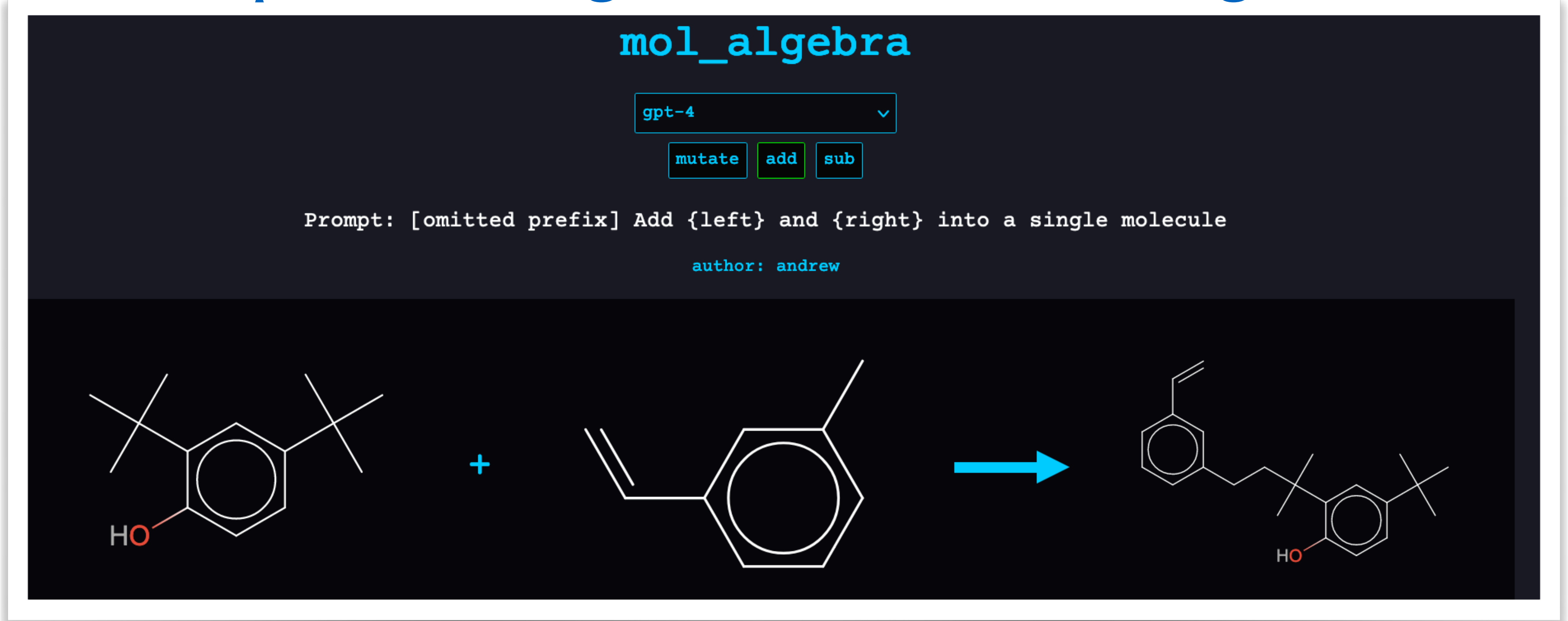


=

"CN1C=NC2=C1C(=O)N(C(=O)N2C)C"

Simplified Molecular-Input Line-Entry System (SMILES)

https://whitead.github.io/svelte-chem-algebra/



Meta AI, Galactica: A Large Language Model for Science, 2211.09085

Modality	Sequence
Text	`Abell 370 is a cluster...`
LaTeX	`r_{s} = \frac{2GM}{c^2}`
Code	`class Transformer(nn.Module)`
SMILES	`C(C(=O)O)N`
AA Sequence	`MIRLGAPQTL..`
DNA Sequence	`CGGTACCCTC..`

Comment | Published: 19 May 2023

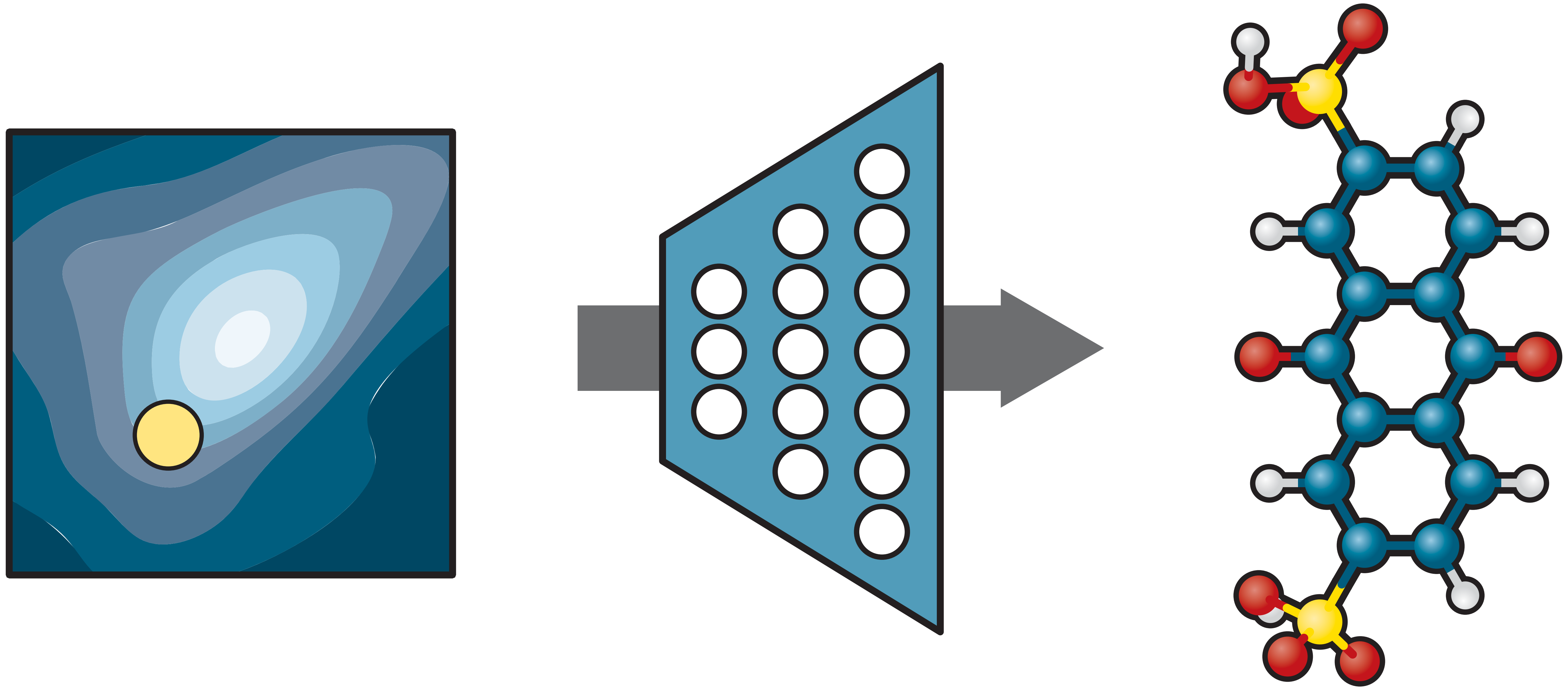
The future of chemistry is language

Andrew D. White

Nature Reviews Chemistry (2023) | Cite this article

Generative AI for matter engineering

latent space

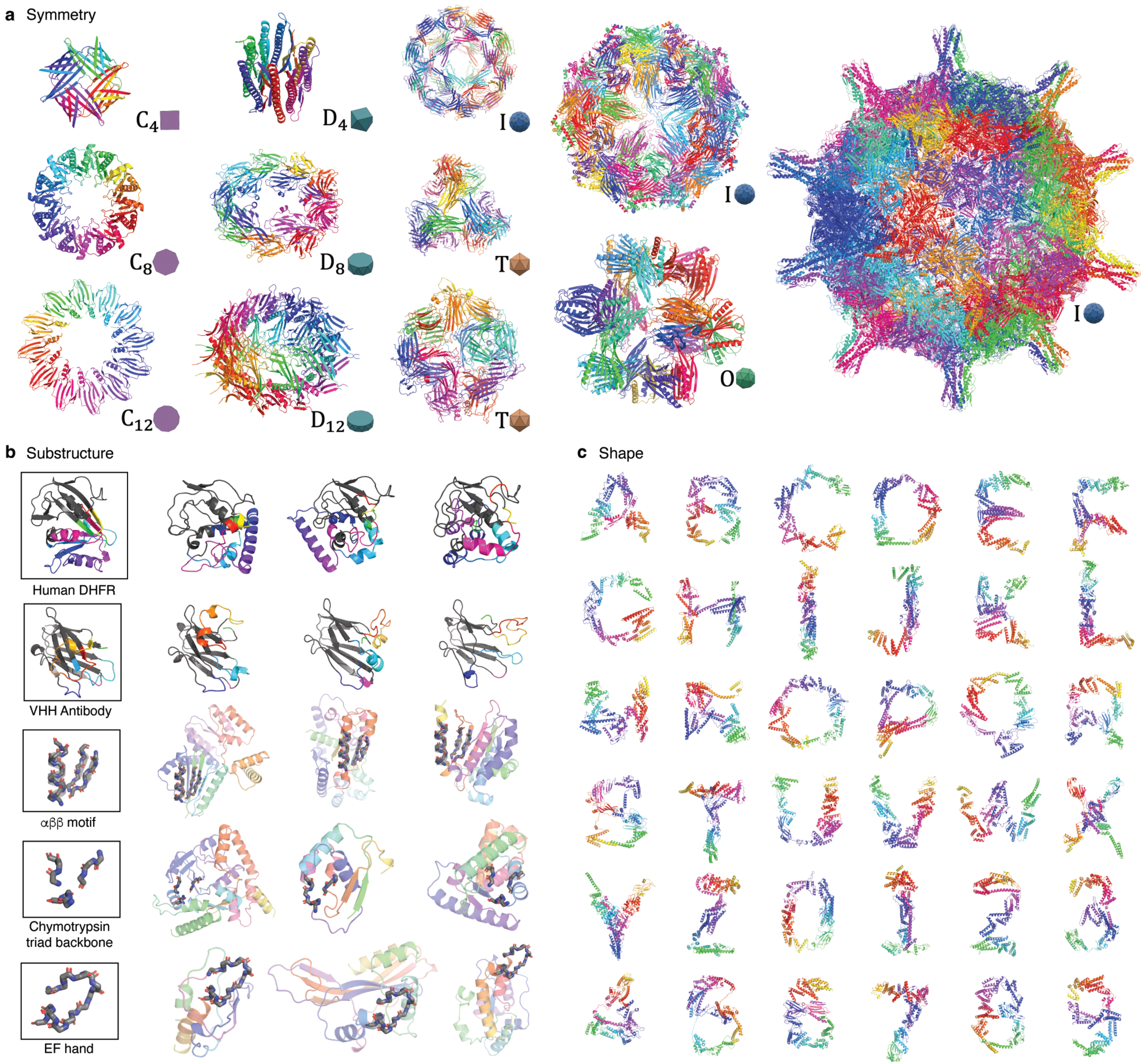


chemical space

Generation $p(\boldsymbol{x} \mid y)$

Prediction $p(y \mid \boldsymbol{x})$

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

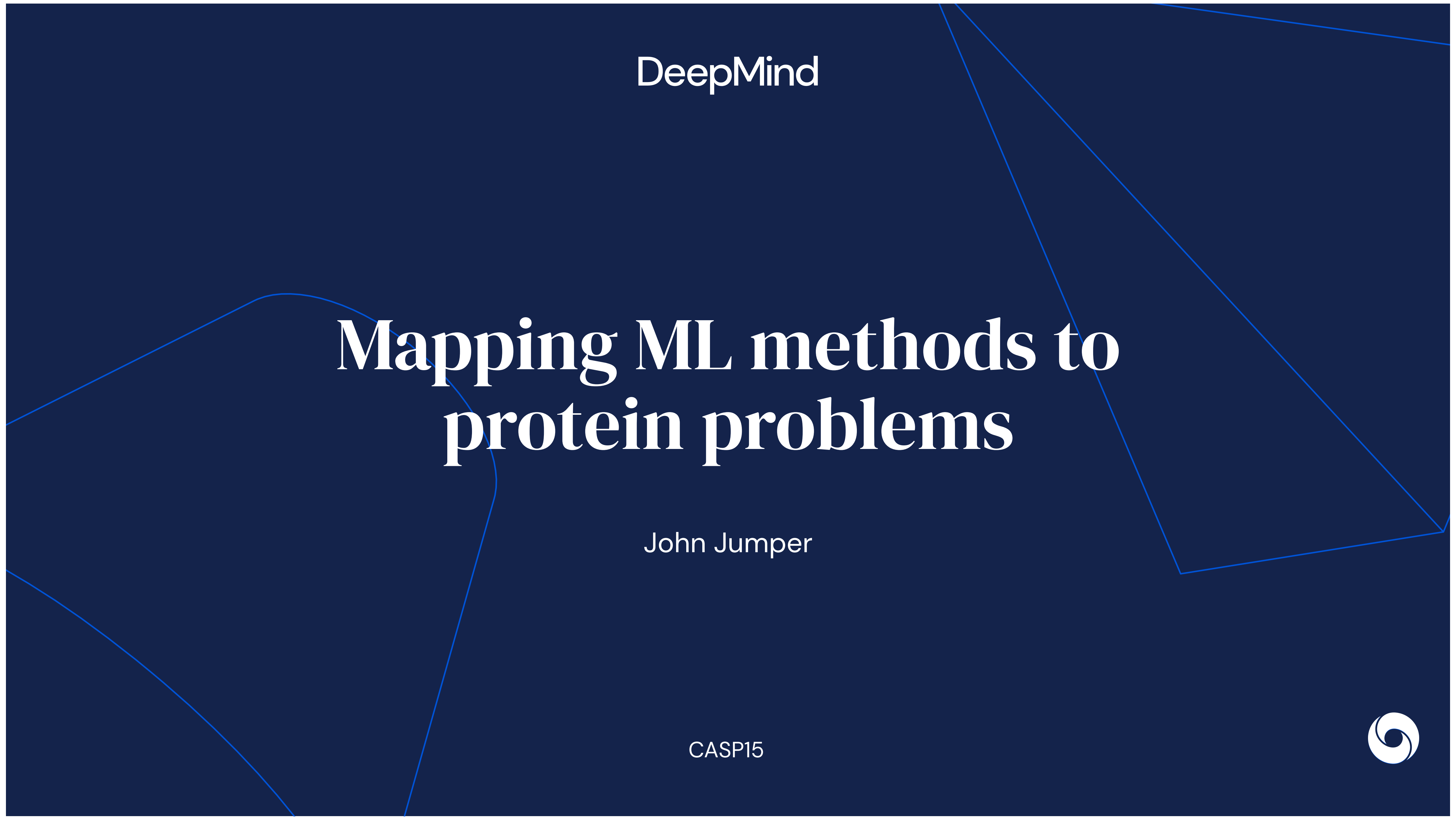


$p(\text{protein} \mid \text{symmetry})$

$p(\text{protein} \mid \text{substructure})$

$p(\text{protein} \mid \text{shape})$

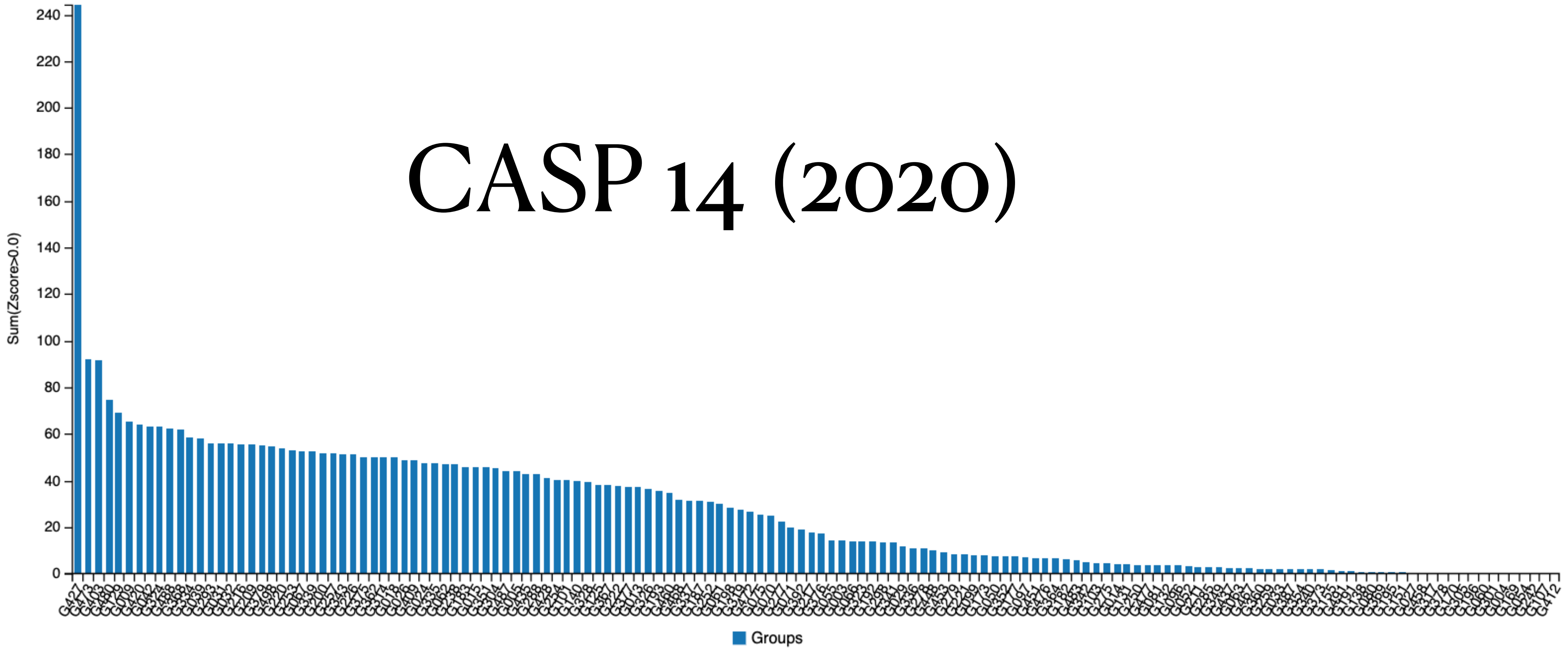
https://generatebiomedicines.com/chroma

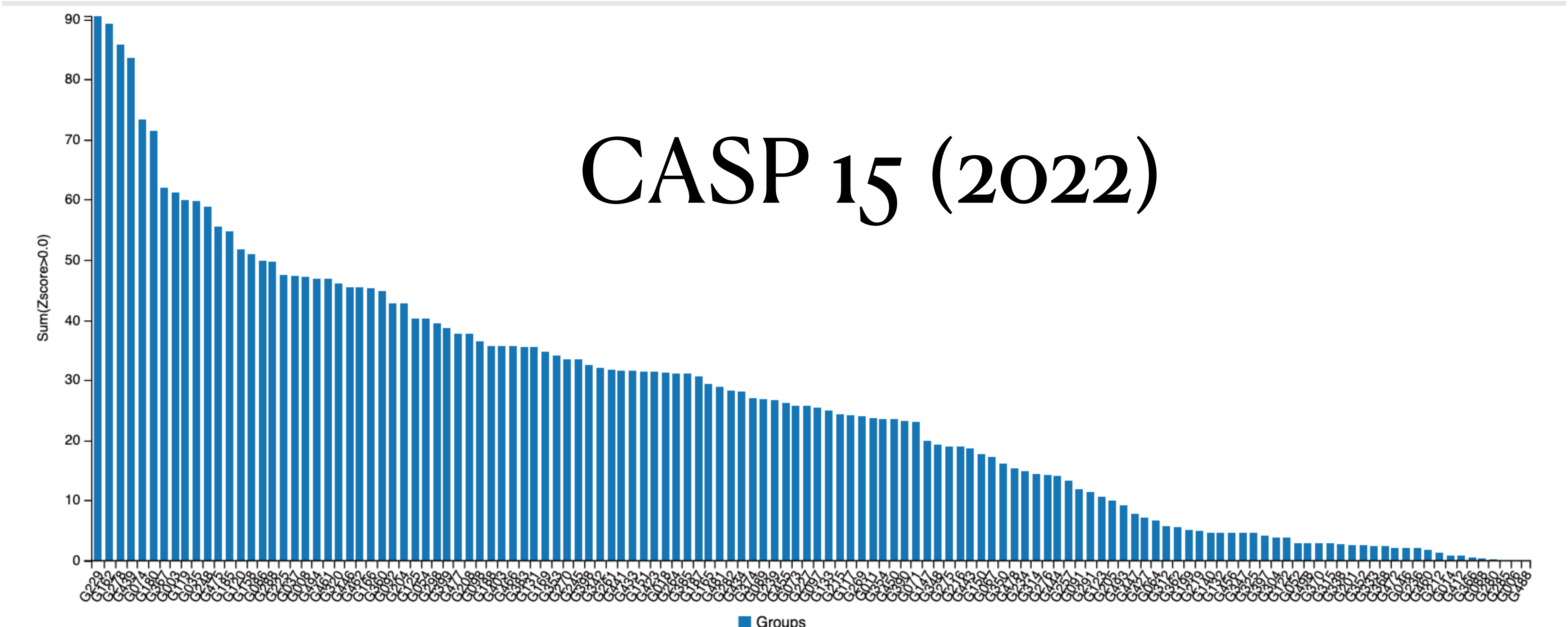


CASP 15 invited talk by John Jumper

Outline

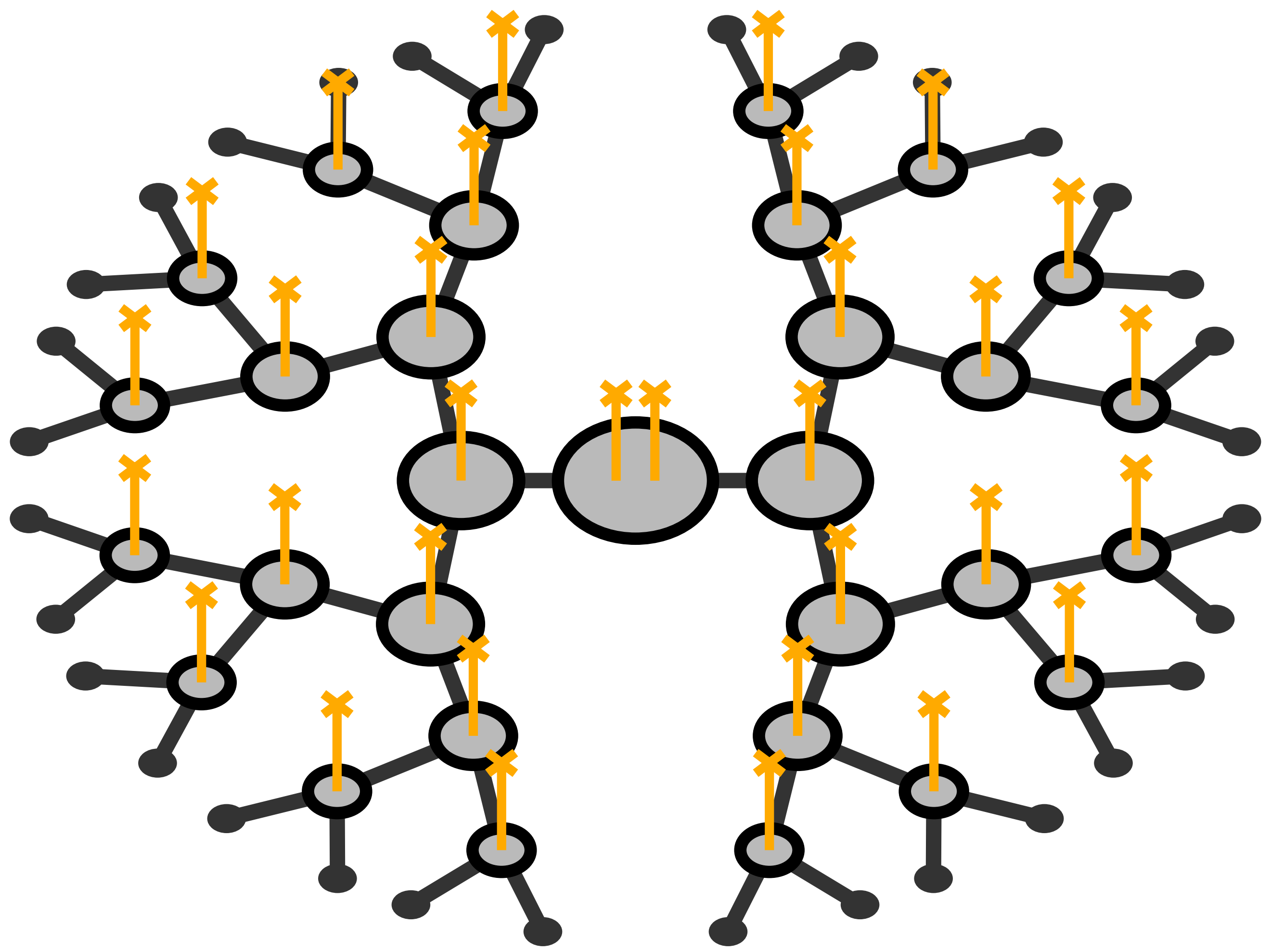
- Generative models and diffusion
- Protein language models and the scaling hypothesis
- Next problems





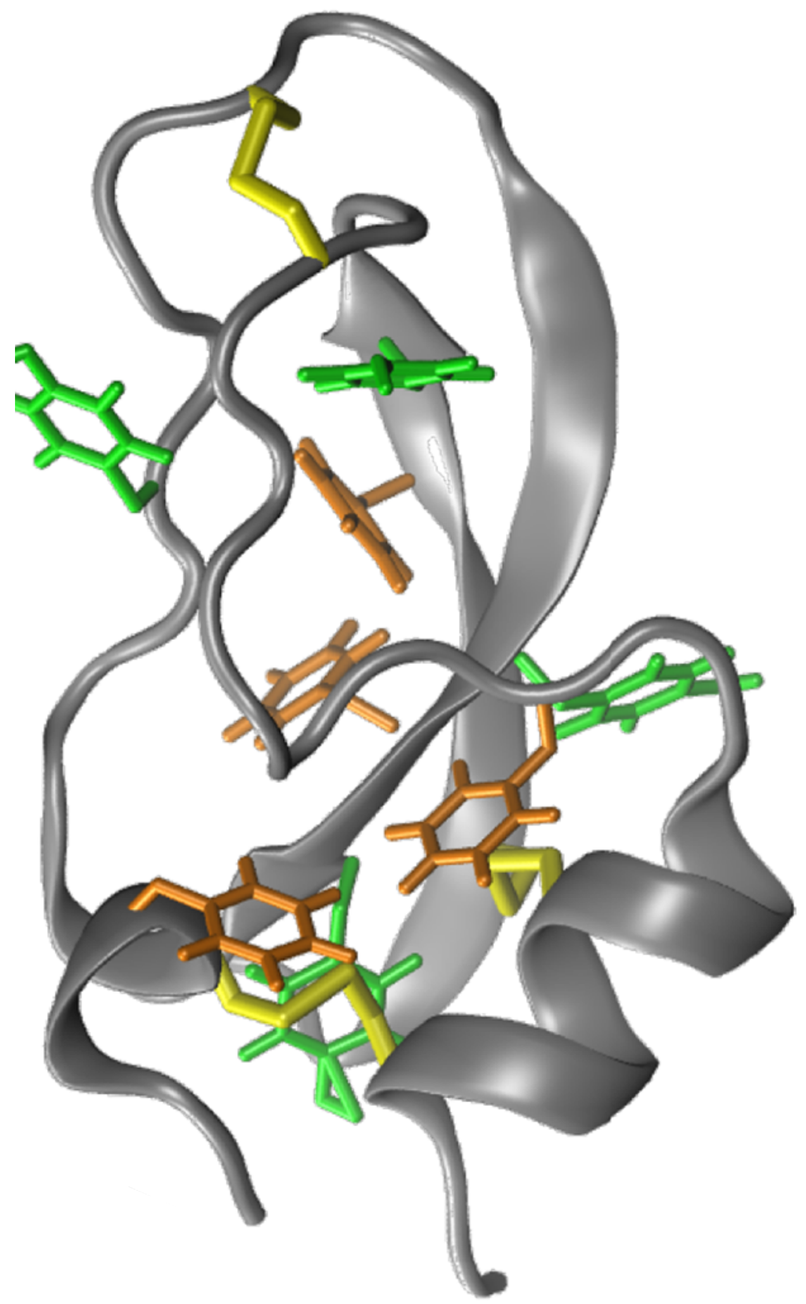
Generative AI for matter computation

Renormalization group



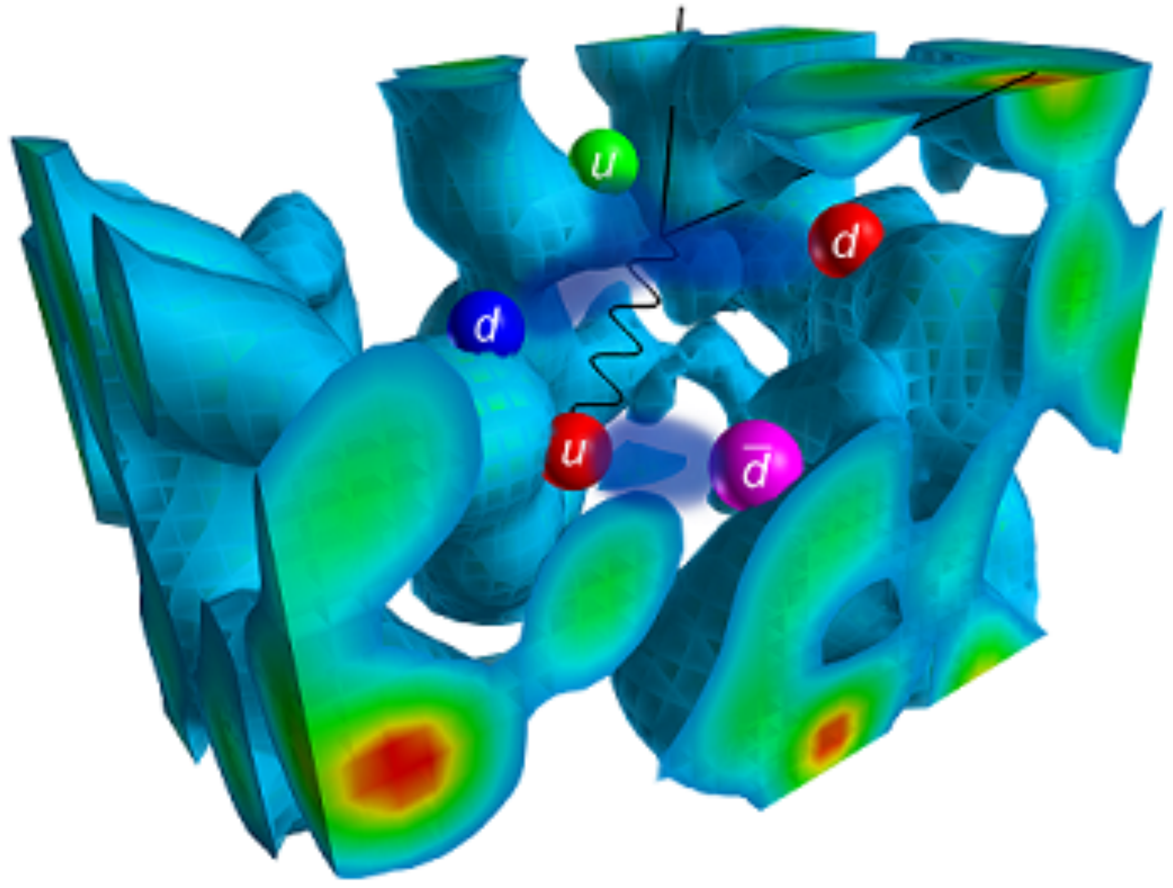
Li and LW, PRL '18
Li, Dong, Zhang, LW, PRX '20

Molecular simulation



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

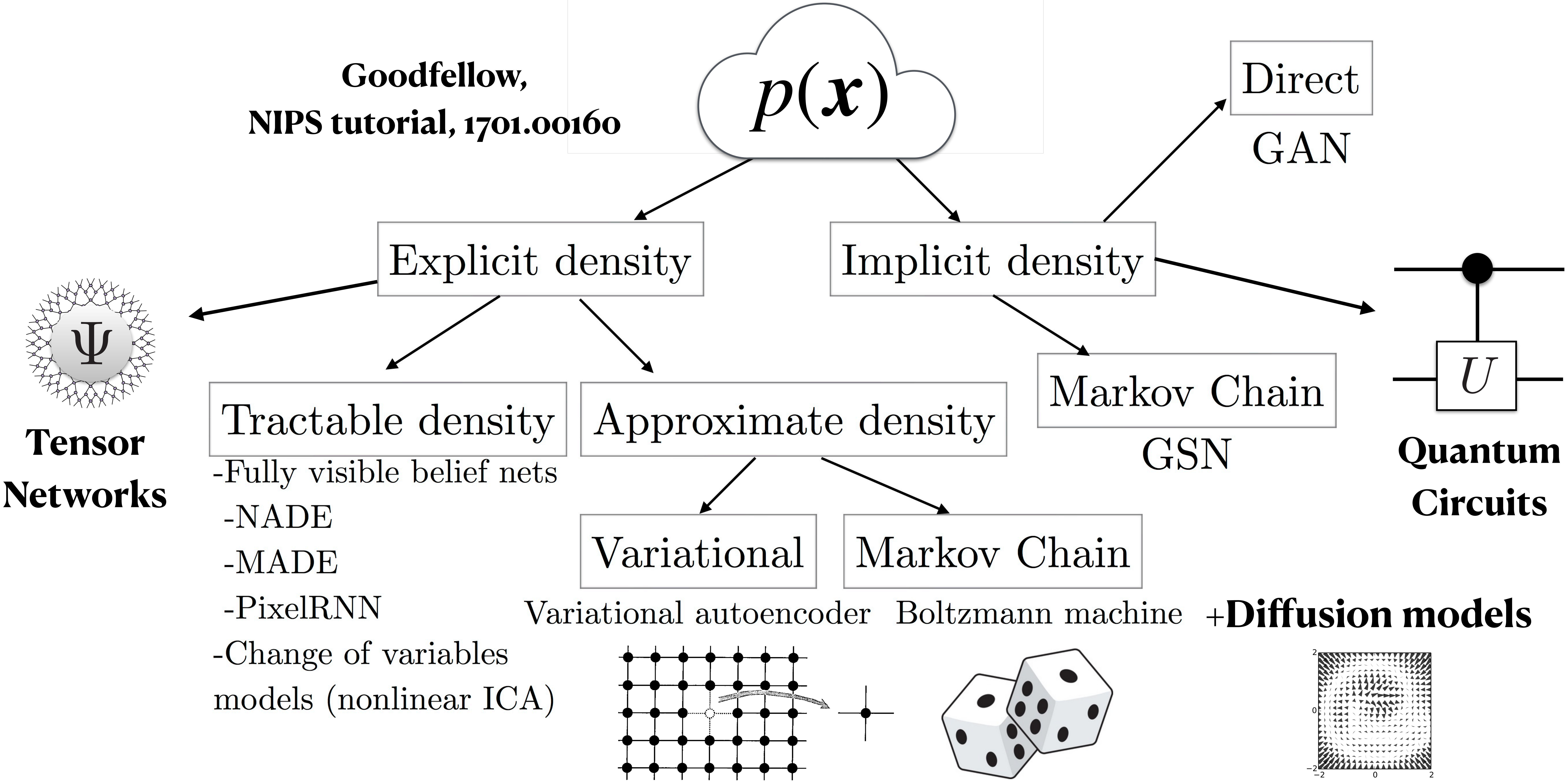
Lattice field theory



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

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

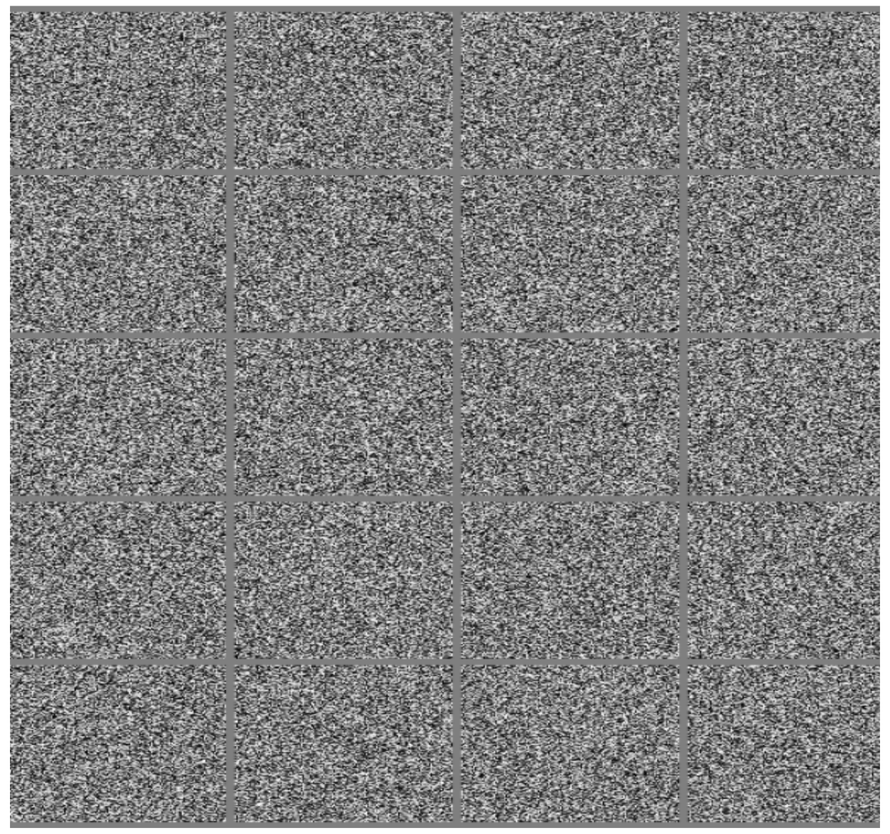
Generative models and their physics genes

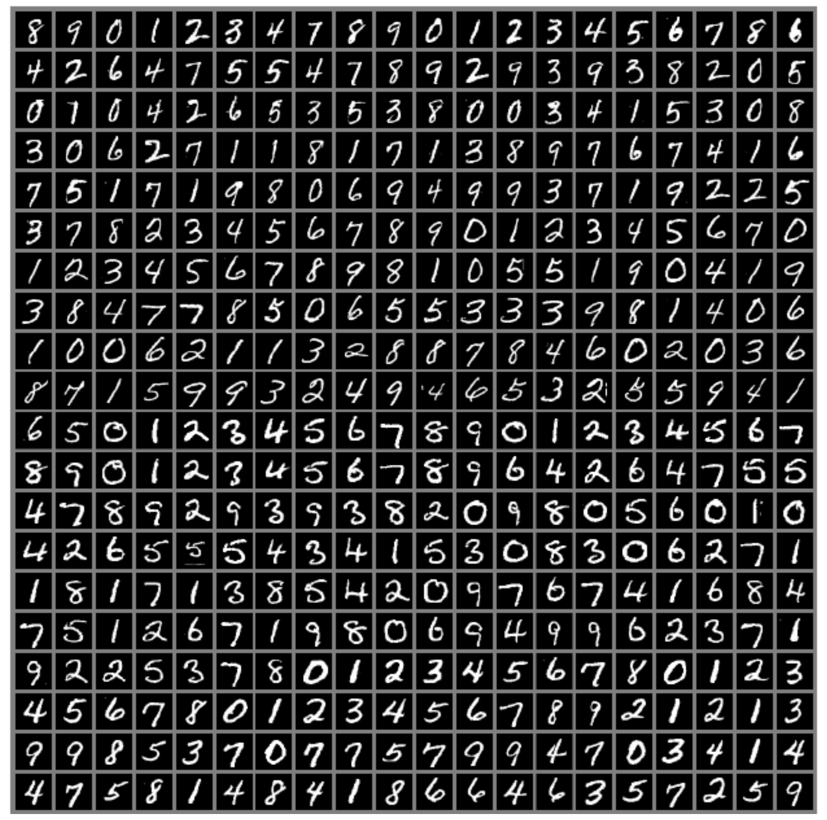


Probabilistic Generative Modeling

$$p(\boldsymbol{x})$$

How to express, learn, and sample from a high-dimensional probability distribution ?



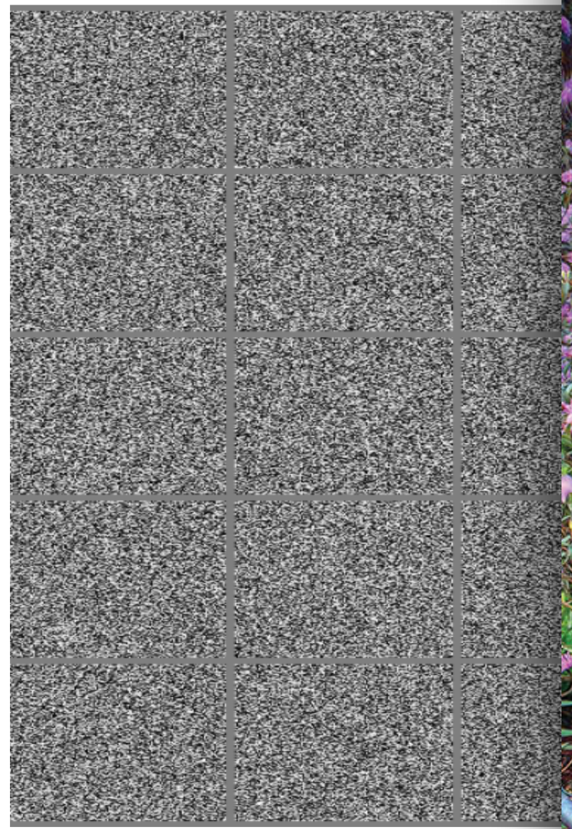


“random” images

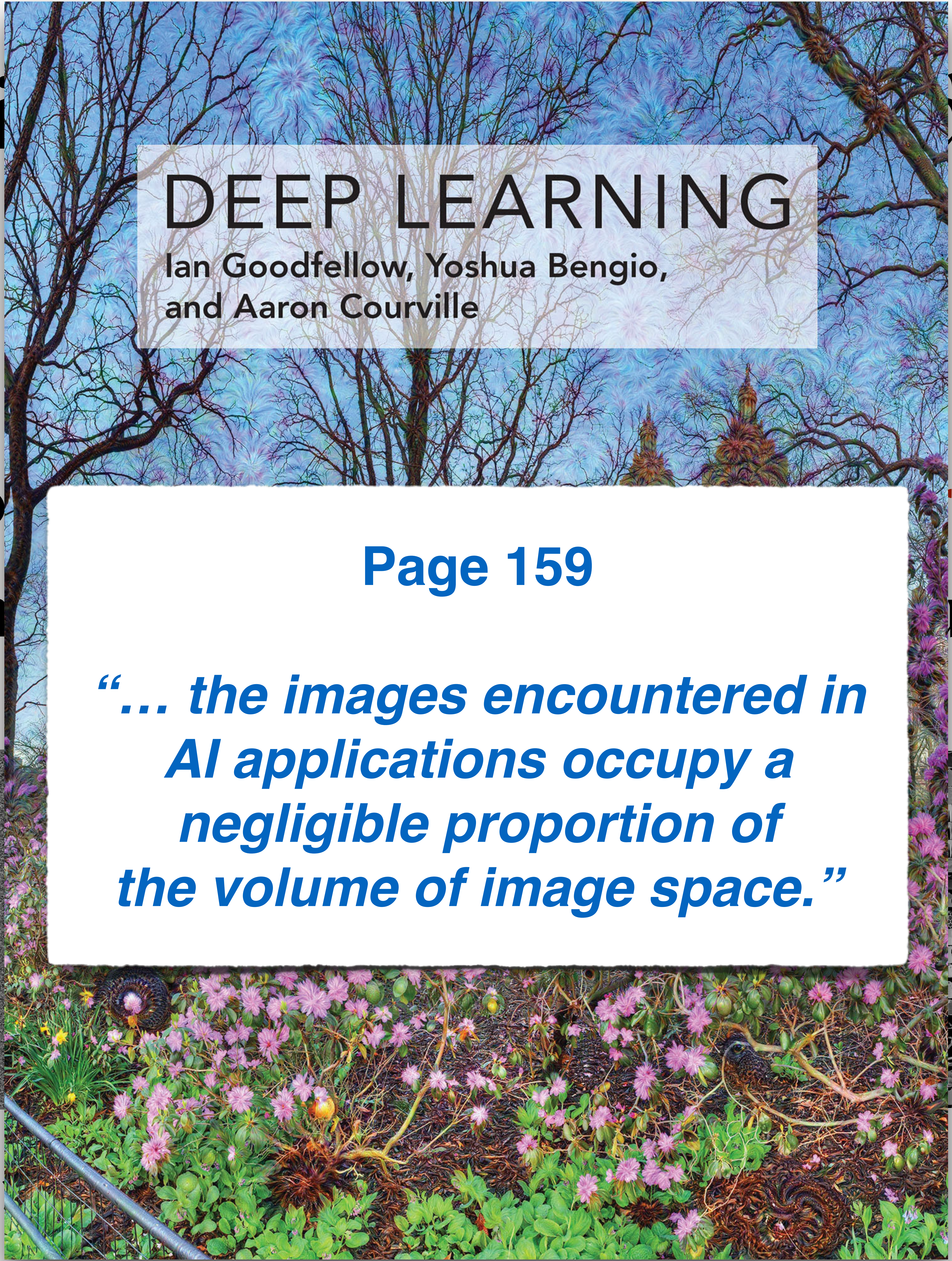
“natural” images

Probabi deling

How to om a
high-di tion ?

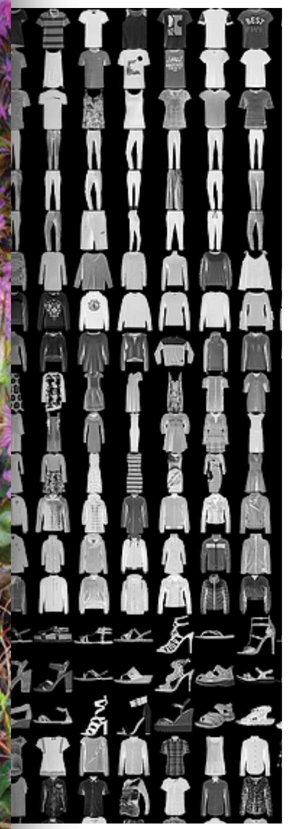


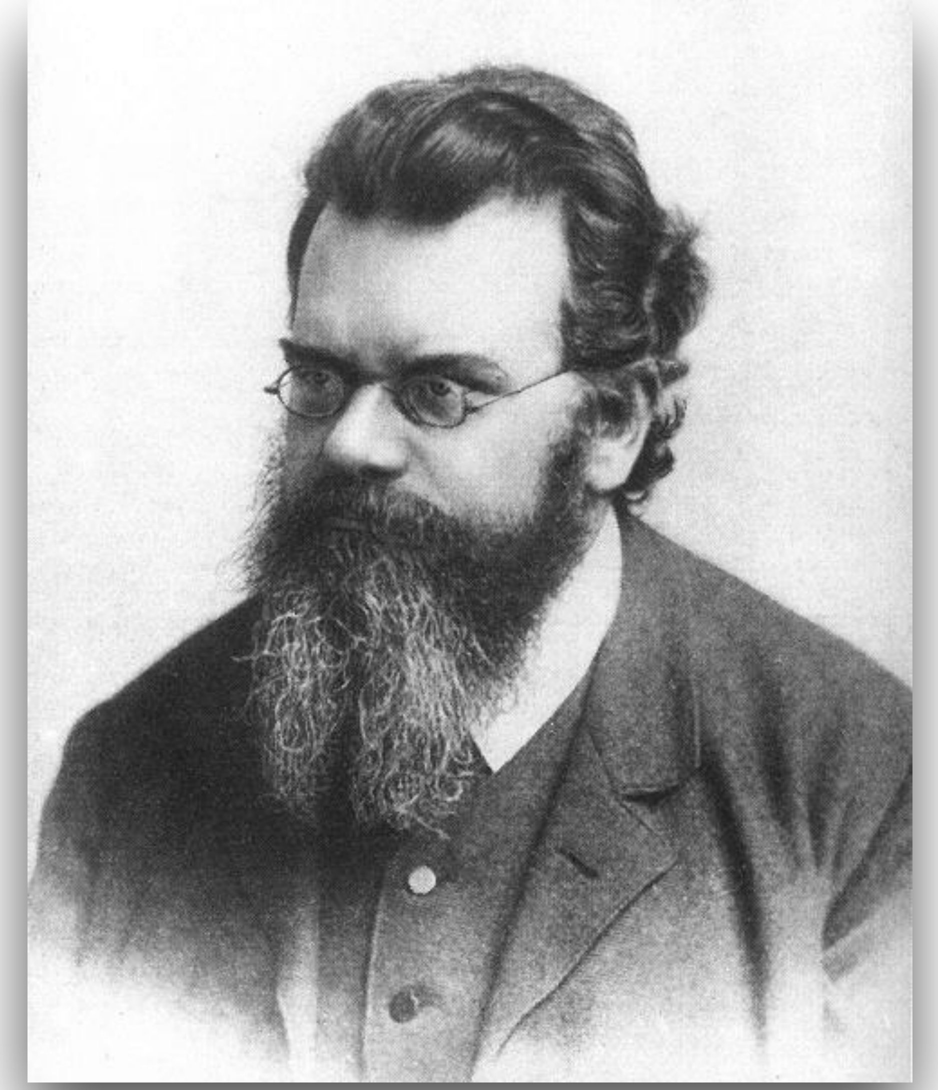
“random”



Page 159

“… the images encountered in AI applications occupy a negligible proportion of the volume of image space.”





Boltzmann Machines

Ackley, Hinton, Sejnowski, Cognitive Science,'85

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

statistical physics

"Born" Machines

Cheng, Chen, LW, Entropy '18,
Han et al, PRX 18', Liu et al PRA '18

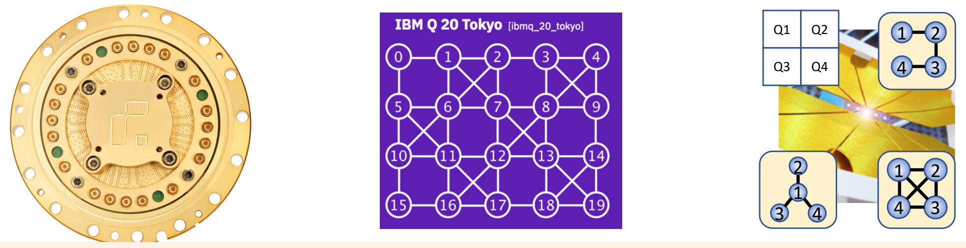
$$p(\boldsymbol{x}) = |\Psi(\boldsymbol{x})|^2$$

quantum physics

Born machine: a quantum (inspired) generative model

$$p(\boldsymbol{x}) = |\Psi(\boldsymbol{x})|^2$$

Quantum circuit realizations



Rigetti to build UK's first commercial quantum computer

Siddharth Venkataramakrishnan in London SEPTEMBER 2 2020

Among the first tasks for the computer is creating a "Quantum Circuit Born Machine", said Alexei Kondratyev, managing director

IonQ and GE R
Potential of Qua
Aggregation

June 23, 2022

COLLEGE PARK, Md.,
promising early results
the benefits of quantum
distributions in risk man

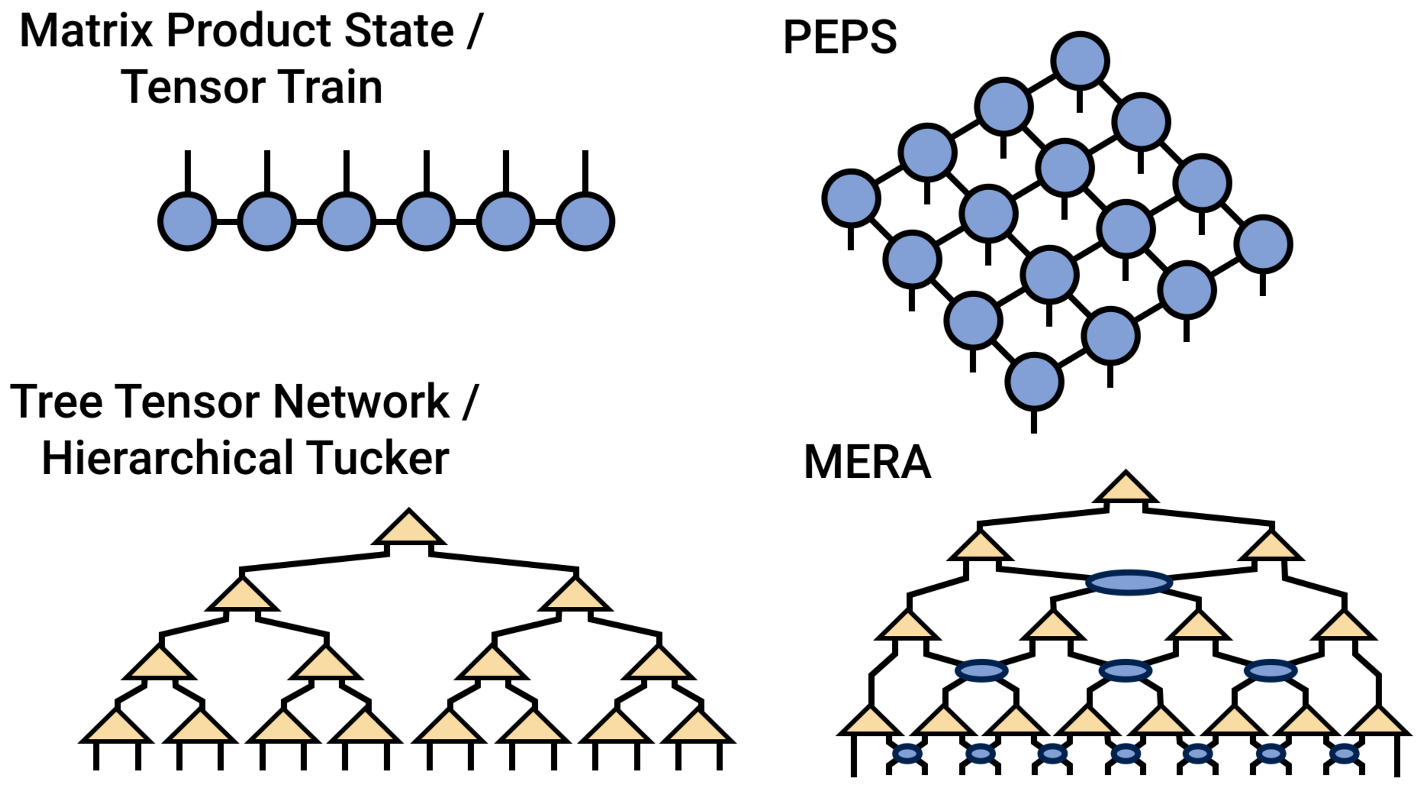
Leveraging a Quantum Circuit Born Machine-based framework on standardized, historical indexes, IonQ and GE Research, the central innovation hub for the G

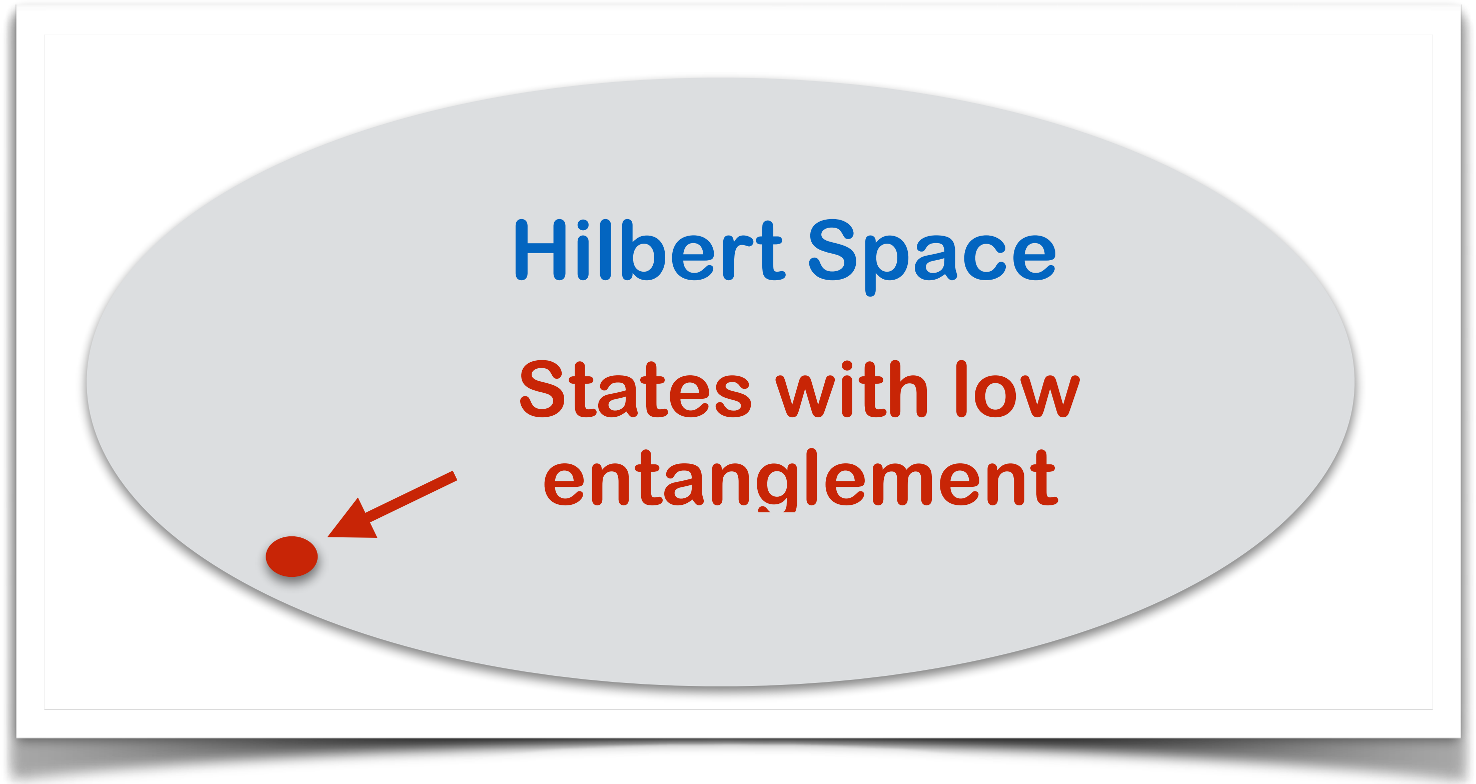
Applications of Quantum Machine Learning

Cambridge Quantum

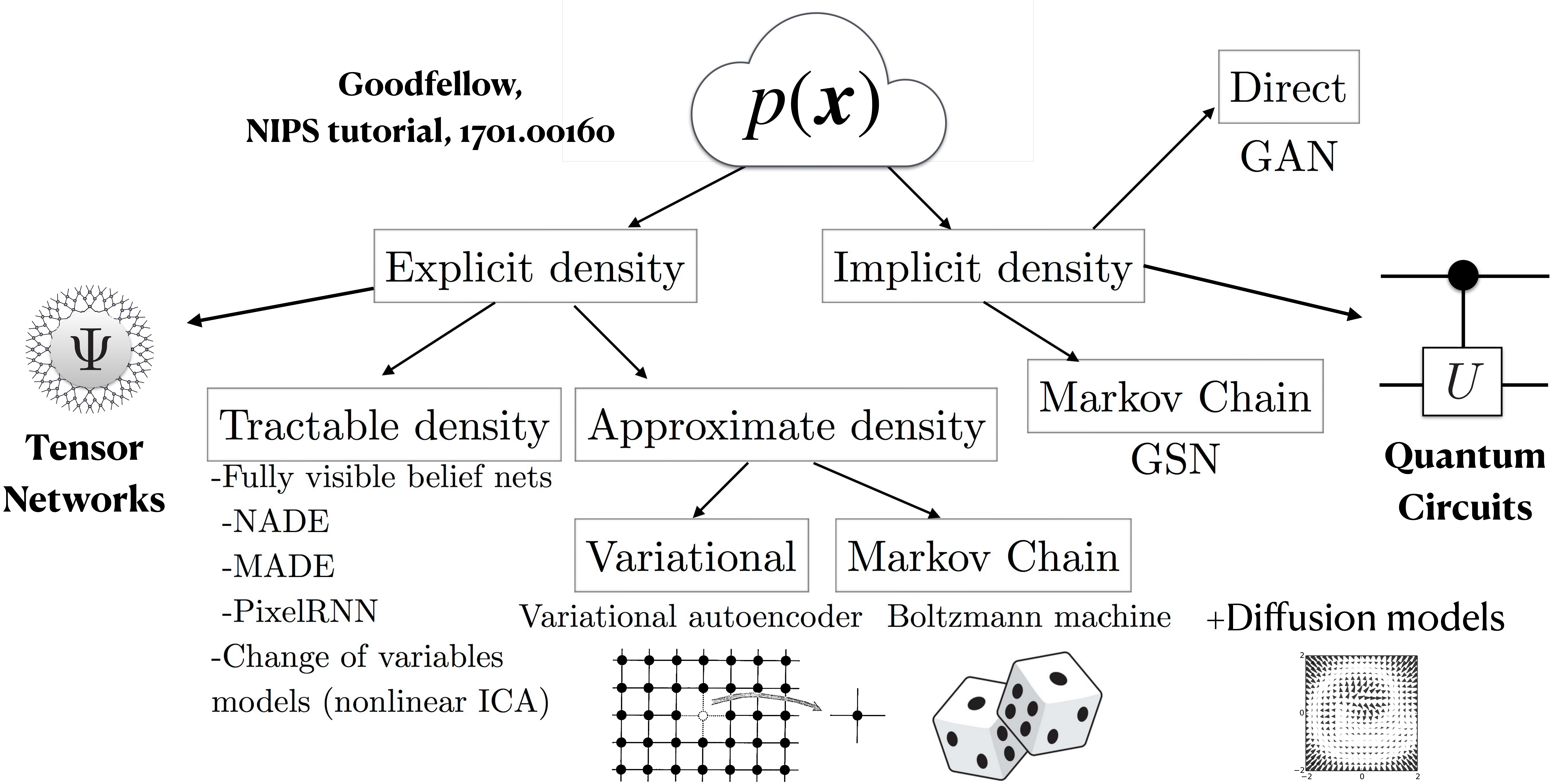
Finance	– Quantum-enhanced variational inference on hidden Markov models for time-series data – Born Machines for foreign exchange spot return modelling – Sampling financial data for Monte Carlo pricing using quantum GANs and Born machines
Pharmaceuticals and Healthcare	– Meta-heuristics for faster biomarker discovery in drug development based on quantum circuit Born machines – Medical diagnosis with quantum-enhanced inference on Bayesian networks

Tensor network Born machines

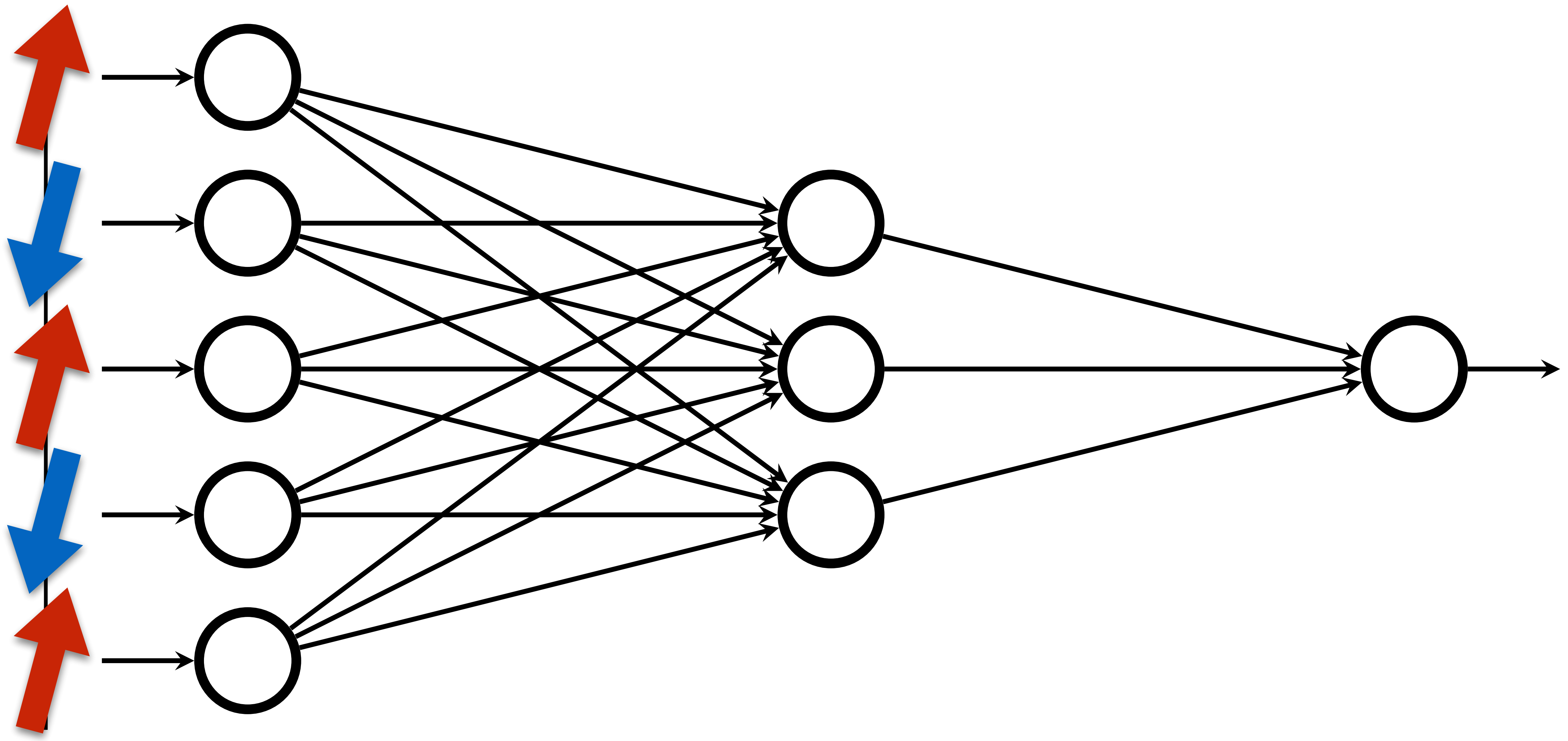




Generative models and their physics genes



So, why bother ?



$\boldsymbol{x}$

$p(\boldsymbol{x}) \geq 0$

Normalization ?

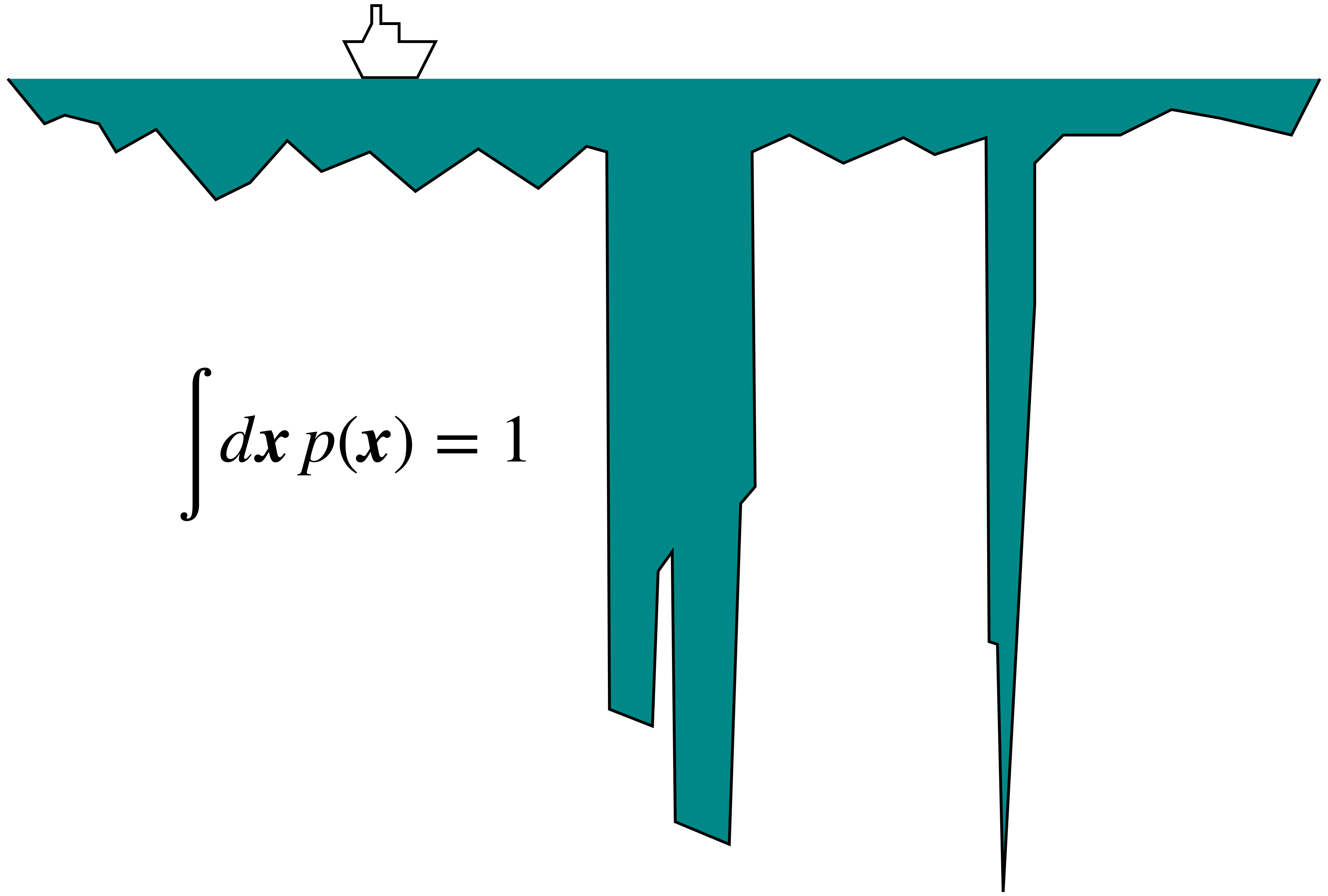
$$\int d\boldsymbol{x}\, p(\boldsymbol{x})$$

Sampling ?

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

So, why bother ?

Normalization

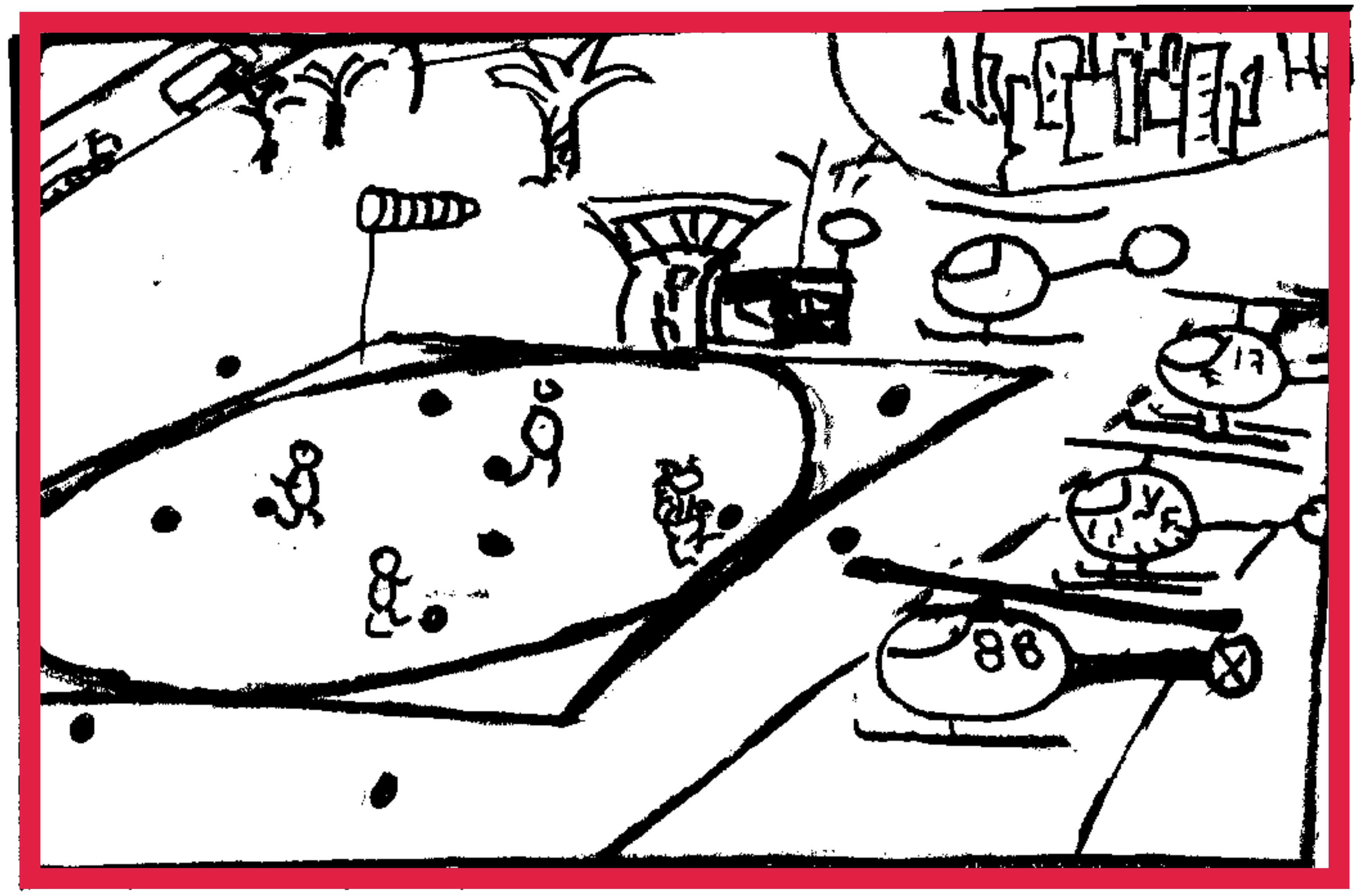


$$\int dx\, p(\boldsymbol{x}) = 1$$

Mackay, Information Theory, Inference, and Learning Algorithms

Sampling

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



Krauth, Statistical Mechanics: Algorithms and Computations

We are going to see how modern generative models resolve these two issues

Generative models	Statistical physics
Negative log-likelihood	Energy function
Score function	Force
Latent variables	Collective variables/coarse graining/renormalization group
Partition function	Free energy calculation
Sample diversity	Enhanced sampling

Two sides of the same coin

Generative modeling

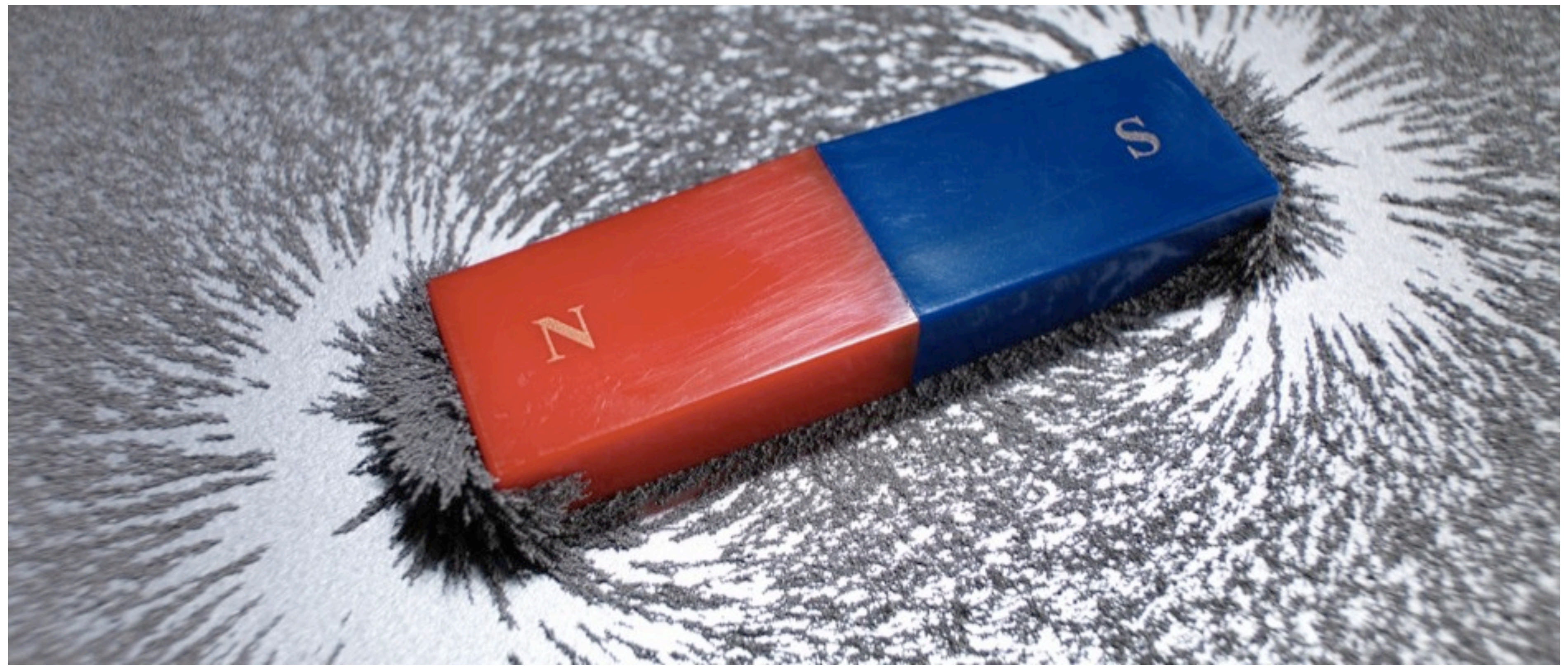
Known: samples

Unknown: generating distribution

“learn from data”

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

Statistical physics



Known: energy function

Unknown: samples, partition function

“learn from energy”

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

$$\mathbb{KL}(\text{data} \parallel p) \quad \text{vs} \quad \mathbb{KL}(p \parallel e^{-E/k_B T})$$

Kullback–Leibler divergence

$$\mathbb{KL}(\pi \parallel p) \equiv \int d\boldsymbol{x}\,\pi(\boldsymbol{x})\left[\ln \pi(\boldsymbol{x}) - \ln p(\boldsymbol{x})\right]$$

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

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

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

Learn from data

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

$$\min_{\theta} \mathbb{KL}(\underset{\text{target}}{\pi} \,\|\, \underset{\text{model}}{p_\theta}) \iff \underbrace{\min_{\theta} \left\{ -\mathbb{E}_{\boldsymbol{x}\sim \text{data}} \left[\ln p_\theta(\boldsymbol{x}) \right] \right\}}_{\text{Maximum likelihood estimation}}$$

The lower bound is the entropy of the dataset: complete memorization

Learn from Energy

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

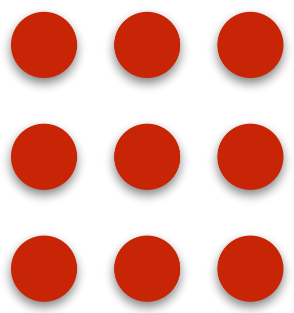
$$\min_{\theta} \mathbb{KL}(\underset{\substack{\uparrow \\ \text{model}}}{p_\theta} \,\|\, \underset{\substack{\uparrow \\ \text{target}}}{\pi}) \iff \underbrace{\min_{\theta} \left\{ \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})} \left[E(\boldsymbol{x}) + k_B T \ln p_\theta(\boldsymbol{x}) \right] \right\}}_{\text{Variational free energy}}$$

The lower bound is the true free energy: exact solution

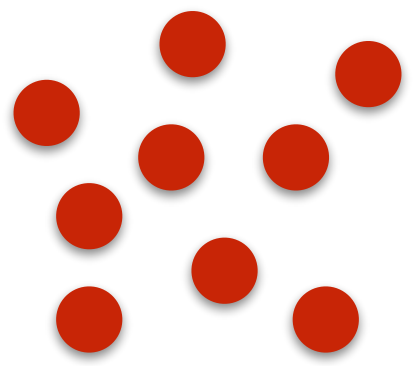
Nature tries to minimize free energy

$$F = E - TS$$

energy



entropy



F is a cost function of Nature

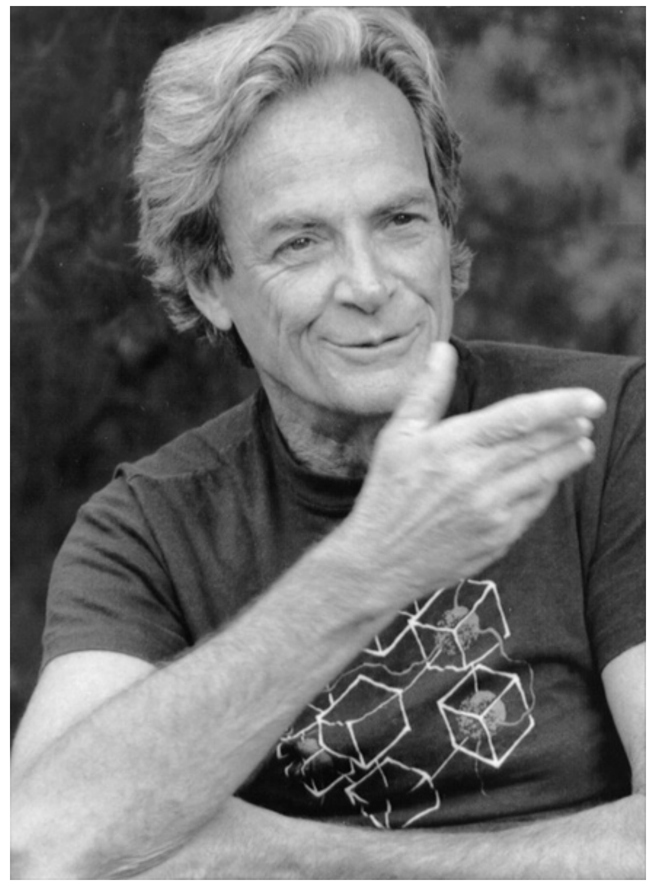
Almost the *same* cost function for training deep generative models

The variational free energy principle

Gibbs–Bogolyubov-Feynman

$$F[p] = \int d\boldsymbol{x}\, p(\boldsymbol{x}) \left[E(\boldsymbol{x}) + k_B T \ln p(\boldsymbol{x}) \right] \geq F$$

variational density ↓ (for $F[p]$), energy ↓ (for $E(\boldsymbol{x})$), entropy ↓ (for $k_B T \ln p(\boldsymbol{x})$)



Difficulties in Applying the Variational Principle to Quantum Field Theories[1]

Richard P. Feynman

Generative models!

[1]transcript of his talk in 1987

Deep variational free energy approach

Deep generative models unlock the power of
the Gibbs–Bogolyubov-Feynman-variational principle

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

$\downarrow$ energy

$\downarrow$ entropy

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

with normalizing flow &
autoregressive models

✓ Tractable entropy

✓ Direct sampling

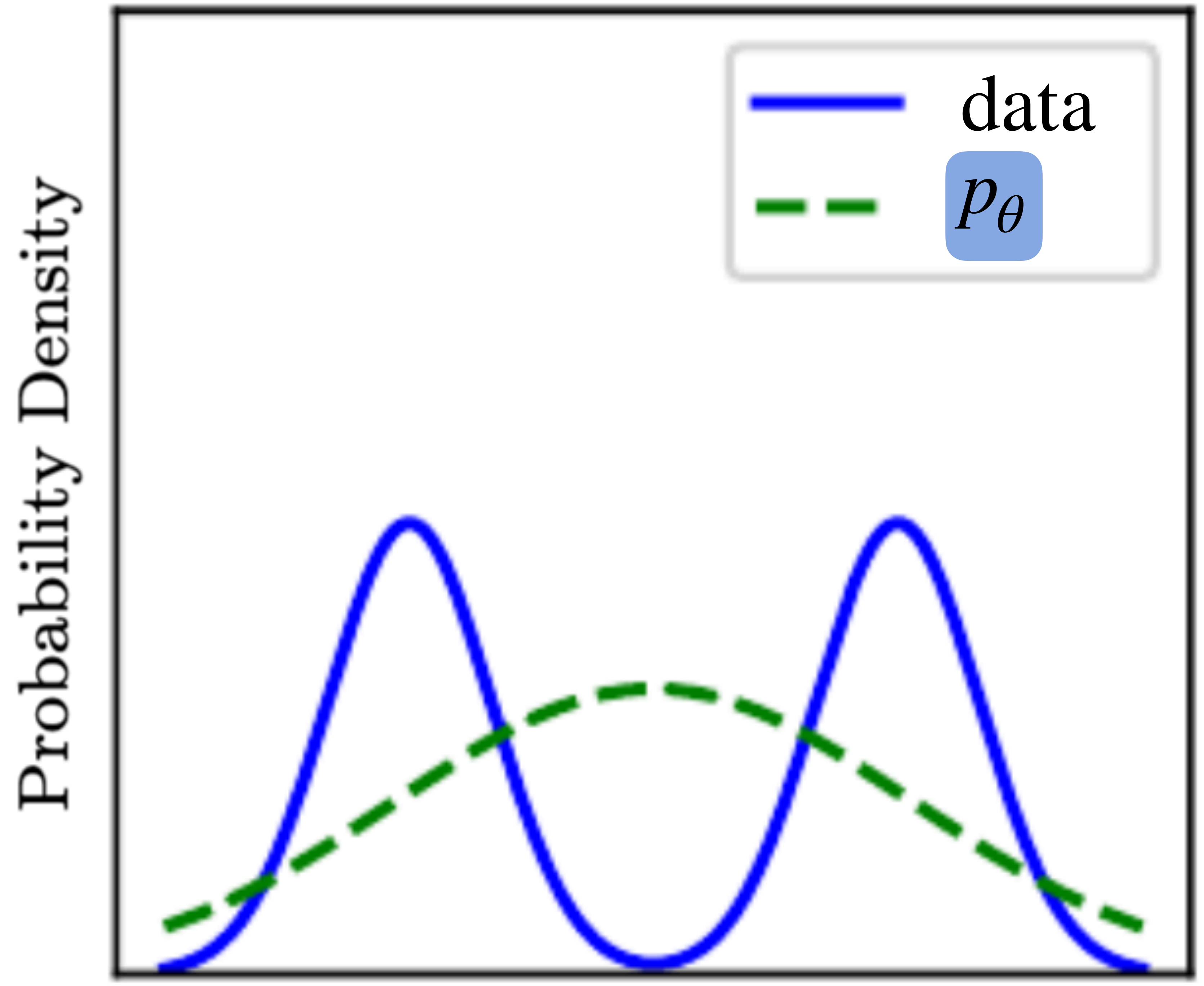
✓ Turning a sampling problem to an optimization problem
better leverages the deep learning engine: 

Forward KL or Reverse KL ?

Maximum likelihood estimation

$$\min_\theta \mathbb{KL}(\text{data} \,\|\, p_\theta)$$

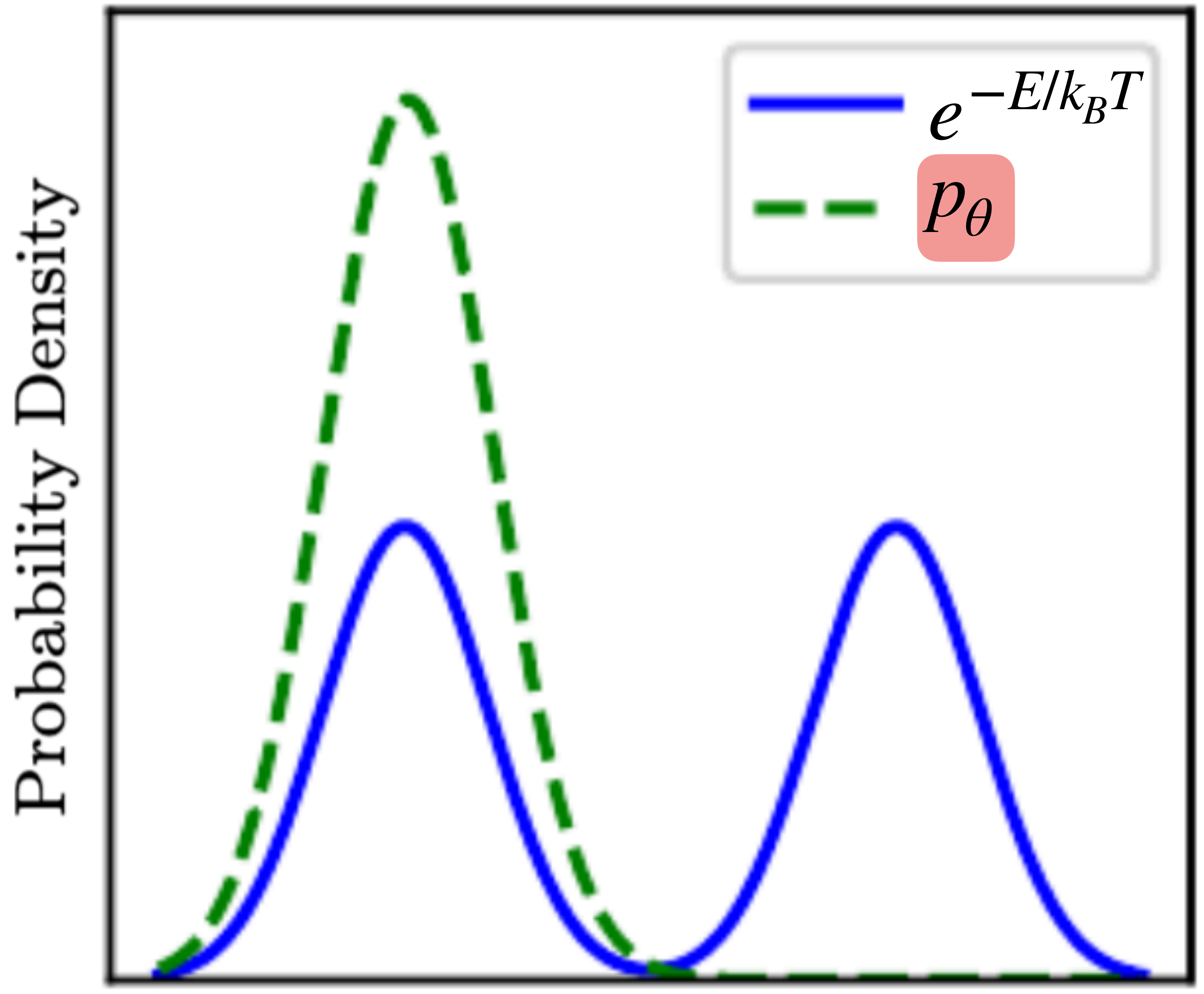
Mode covering



Variational free energy

$$\min_\theta \mathbb{KL}(p_\theta \,\|\, e^{-E/k_BT})$$

Mode seeking



Goodfellow et al, Deep Learning

“Jack of all trades, master of none” — 2302.10724

filling the gap vs pushing the boundary of human knowledge

α-divergence

Minka, Microsoft Research Technical Report 2005

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

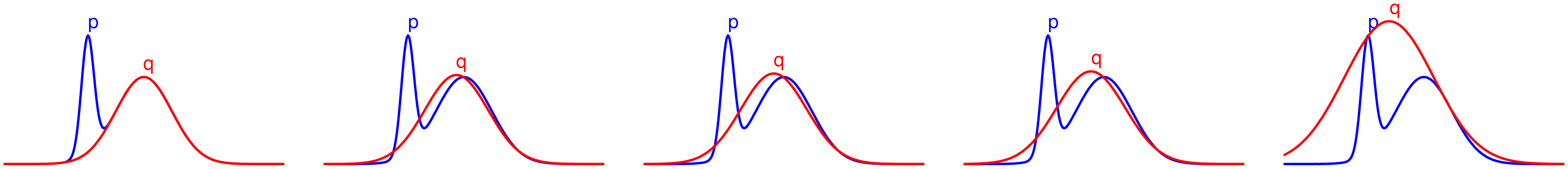
$$D_{-1}(p\,||\,q) = \frac{1}{2}\int_x \frac{(q(x)-p(x))^2}{p(x)}dx$$

$$\lim_{\alpha\to 0} D_\alpha(p\,||\,q) = \mathrm{KL}(q\,||\,p)$$

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

$$\lim_{\alpha\to 1} D_\alpha(p\,||\,q) = \mathrm{KL}(p\,||\,q)$$

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



$\alpha = -\infty$ $\quad$ $\alpha = 0$ $\quad$ $\alpha = 0.5$ $\quad$ $\alpha = 1$ $\quad$ $\alpha = \infty$

Fisher divergence, defined as

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

Autoregressive models

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

Language: GPT 2005.14165

"... quick brown fox *jumps* *..."*

$p(\textcolor{crimson}{jumps}\,|\,\ldots)$

Speech: WaveNet 1609.03499

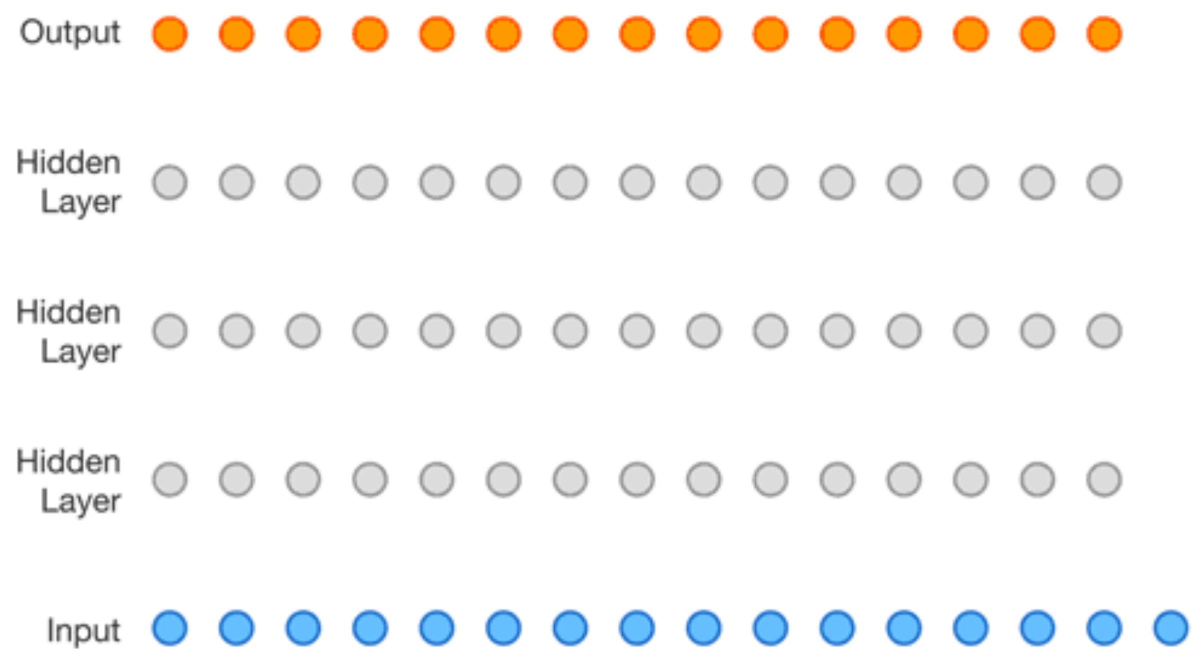
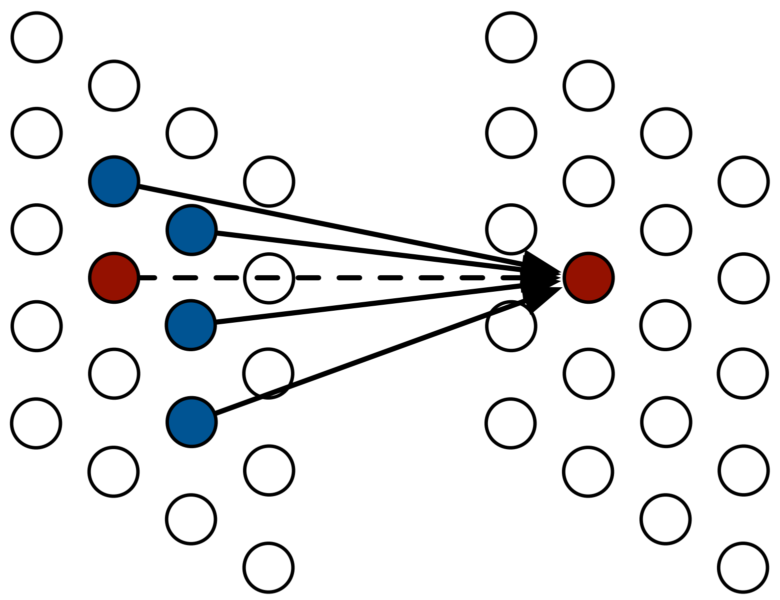
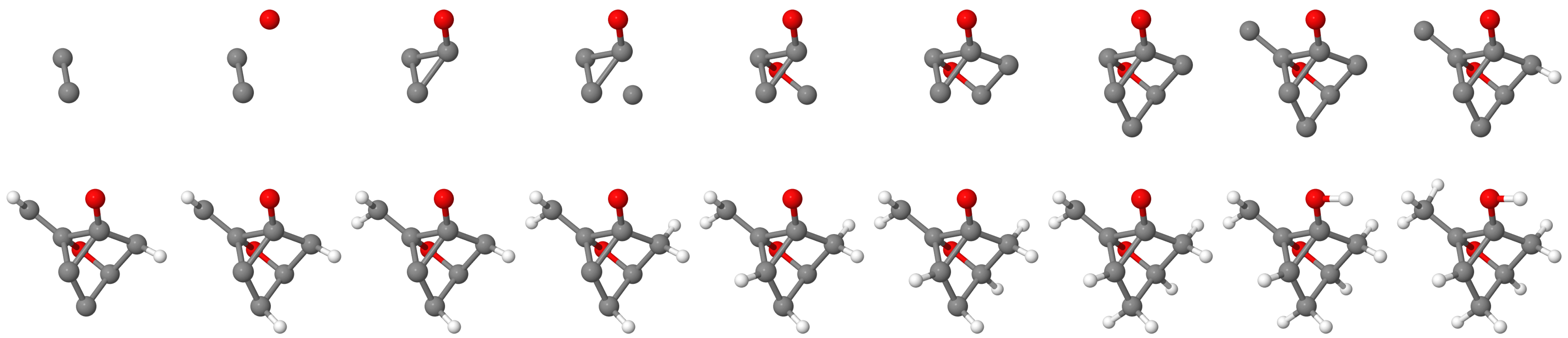


Image: PixelCNN 1601.06759



Molecular graph: 1810.11347



Autoregressive models

$$p(\boldsymbol{x}) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)\cdots$$

Language: GPT 2005.14165

Speech: WaveNet 1609.03499

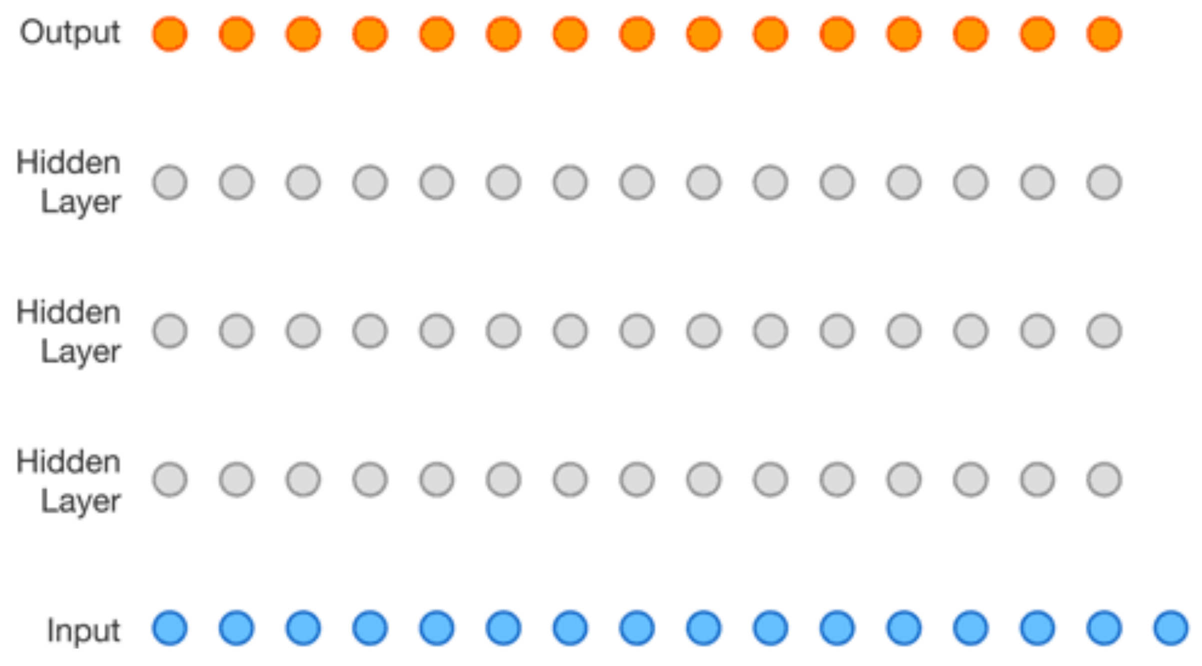
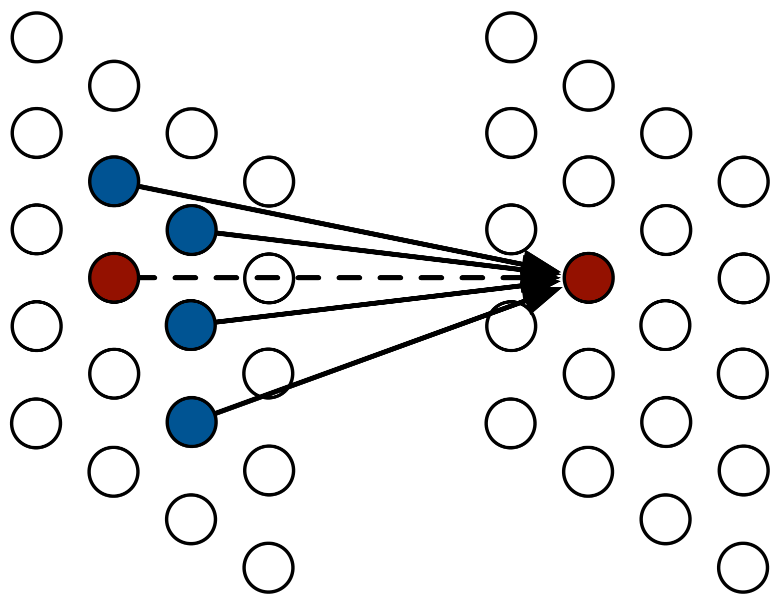
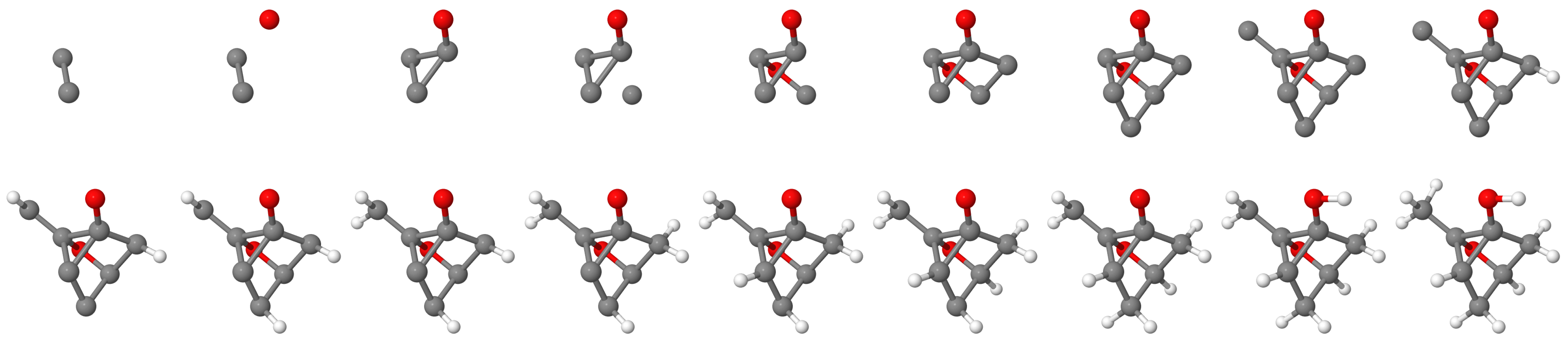


Image: PixelCNN 1601.06759

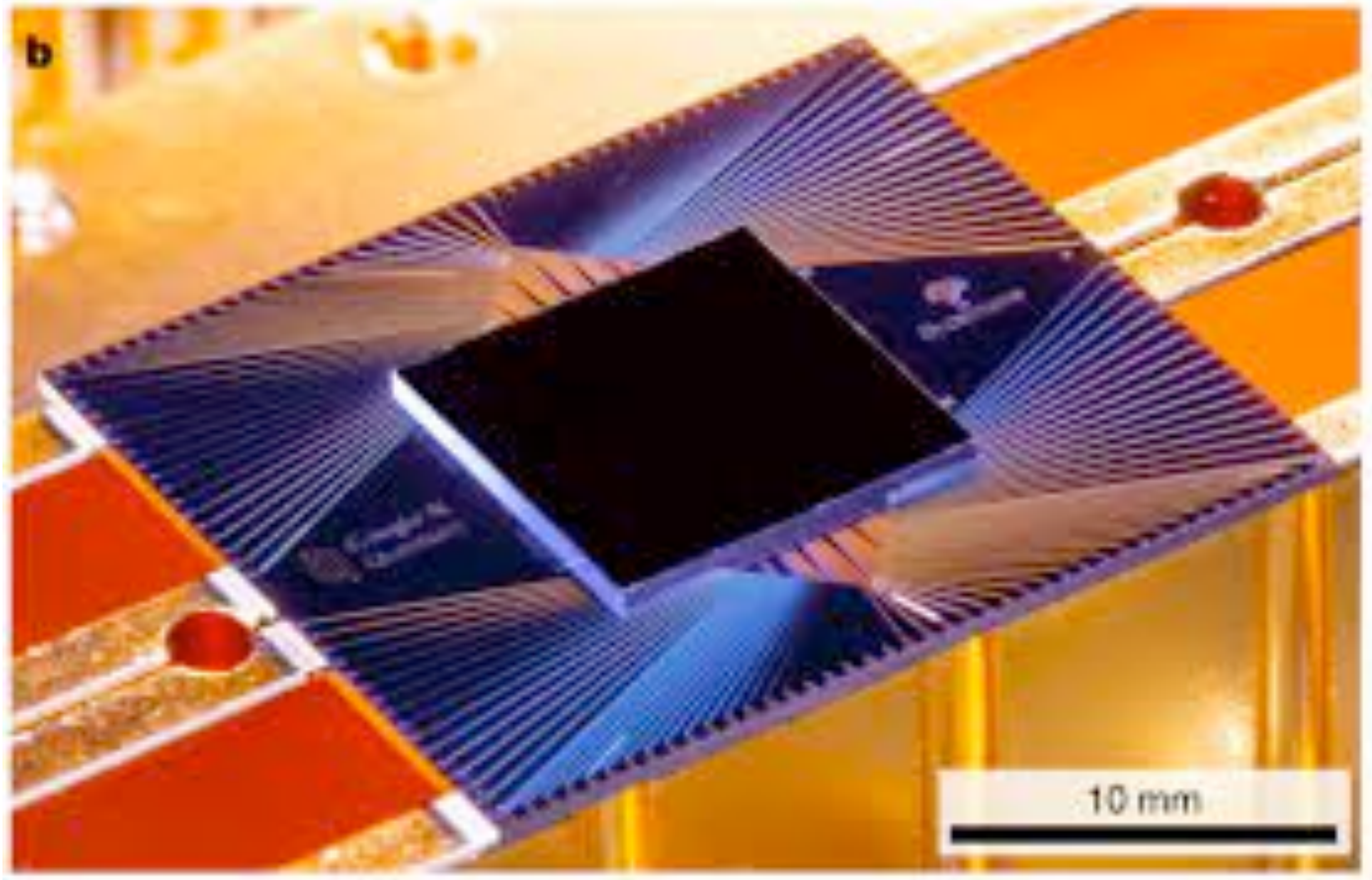


Molecular graph: 1810.11347



Demo: Generative model of Sycamore data

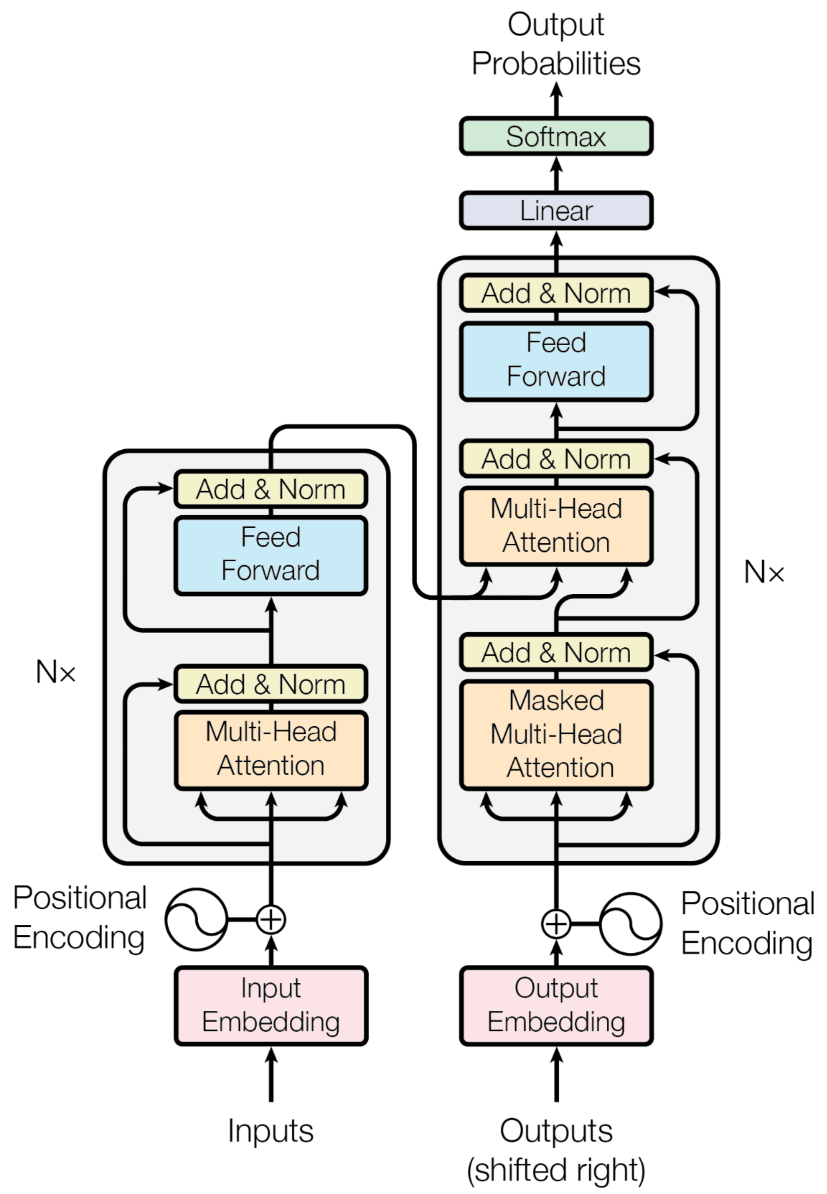
Quantum chip



bitstrings $\sim |\Psi(\boldsymbol{x})|^2$

```
011110110100
100001111011
100110110111
100110100010
010100011000
010001000000
010101101100
100001111000
100101001001
001000001010
```

Transformer

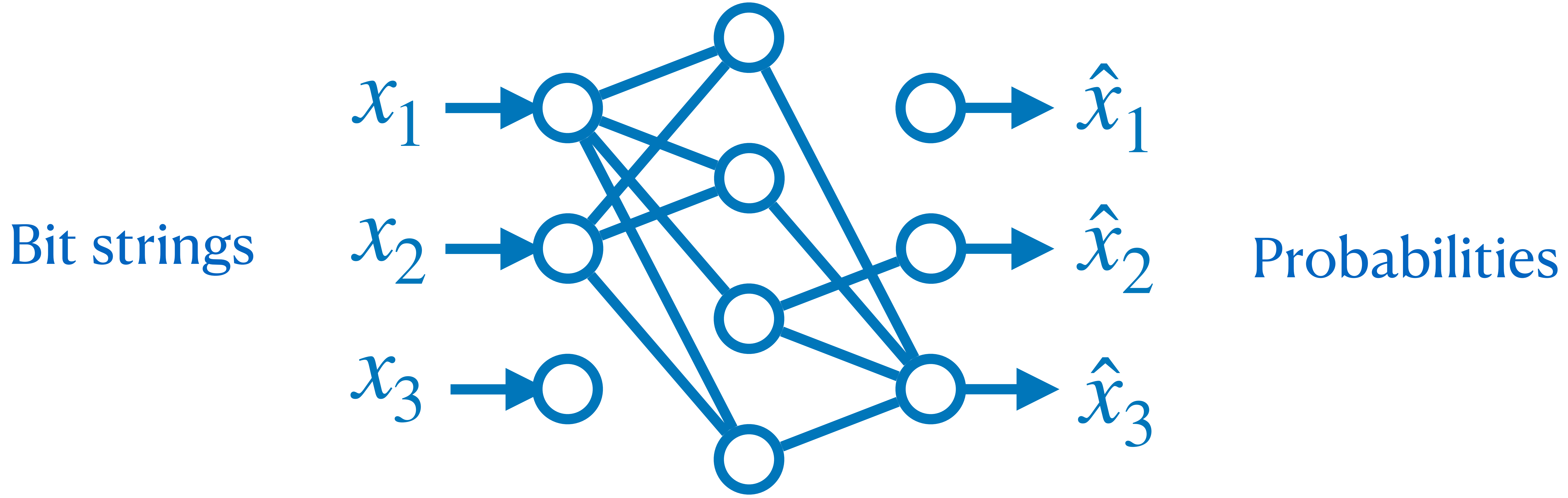


Can we fake the measurement of the sycamore quantum circuit by training a transformer?

https://colab.research.google.com/drive/11WaroqULkudKT3h2i5J6r_EmA4wFKkoZ?usp=sharing

Implementation: autoregressive masks

Masked Autoencoder Germain et al, 1502.03509



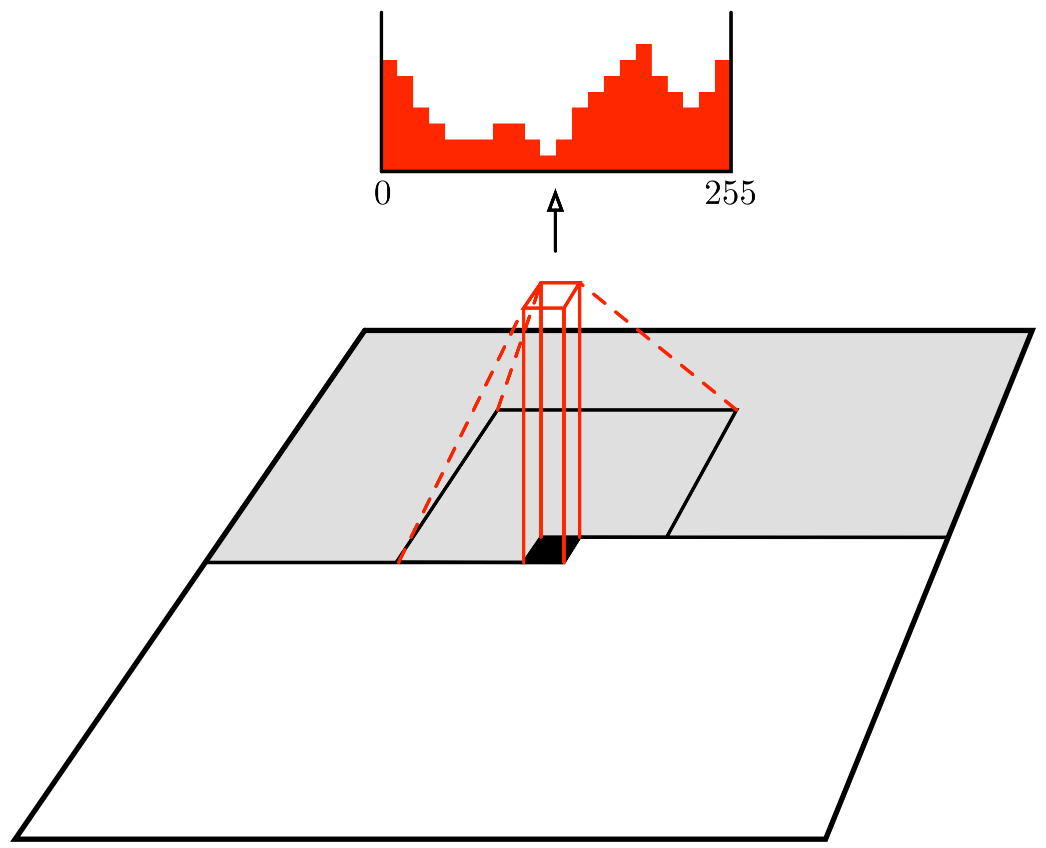
$p(x_1) = \text{Bernoulli}(\hat{x}_1)$ $\quad p(x_2 \mid x_1) = \text{Bernoulli}(\hat{x}_2)$ $\quad p(x_3 \mid x_1, x_2) = \text{Bernoulli}(\hat{x}_3)$

Other ways to implement autoregressive models: recurrent networks

Implementation: autoregressive masks

Masked convolutional

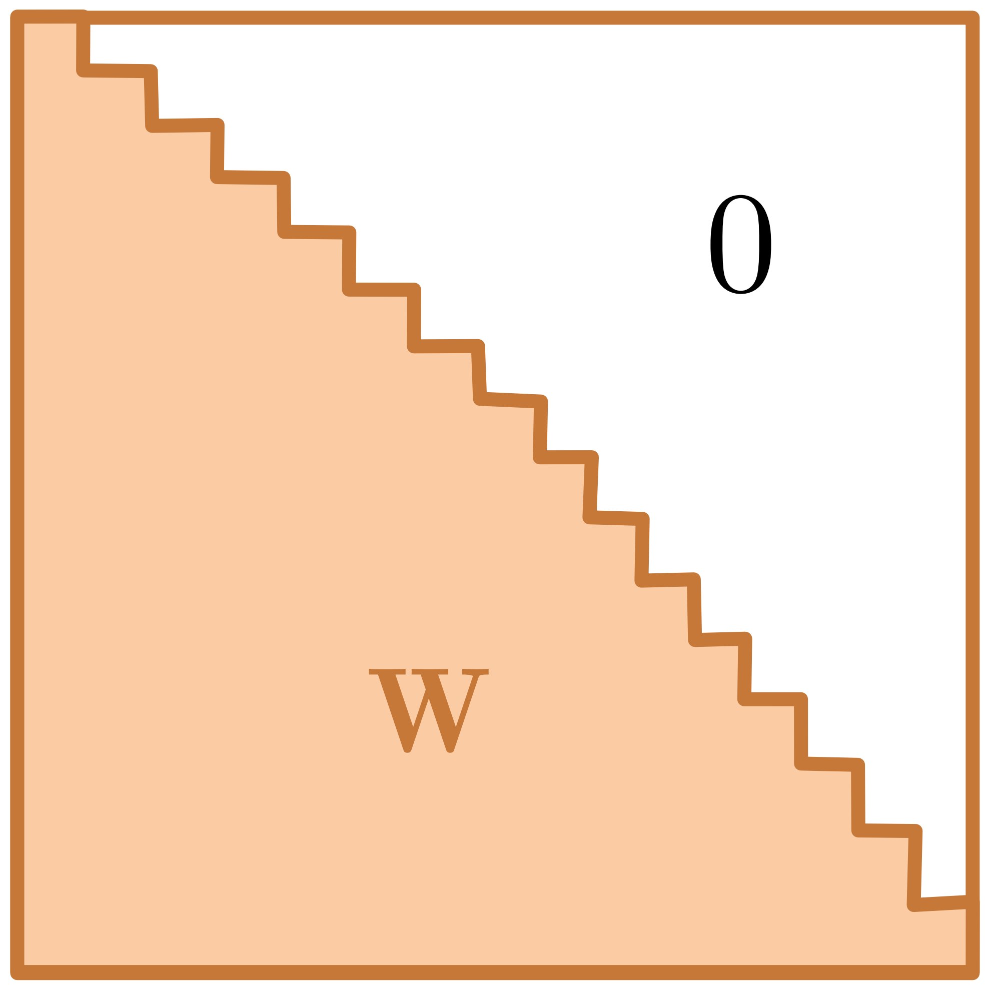
PixelCNN, van den Oord et al, 1601.06759



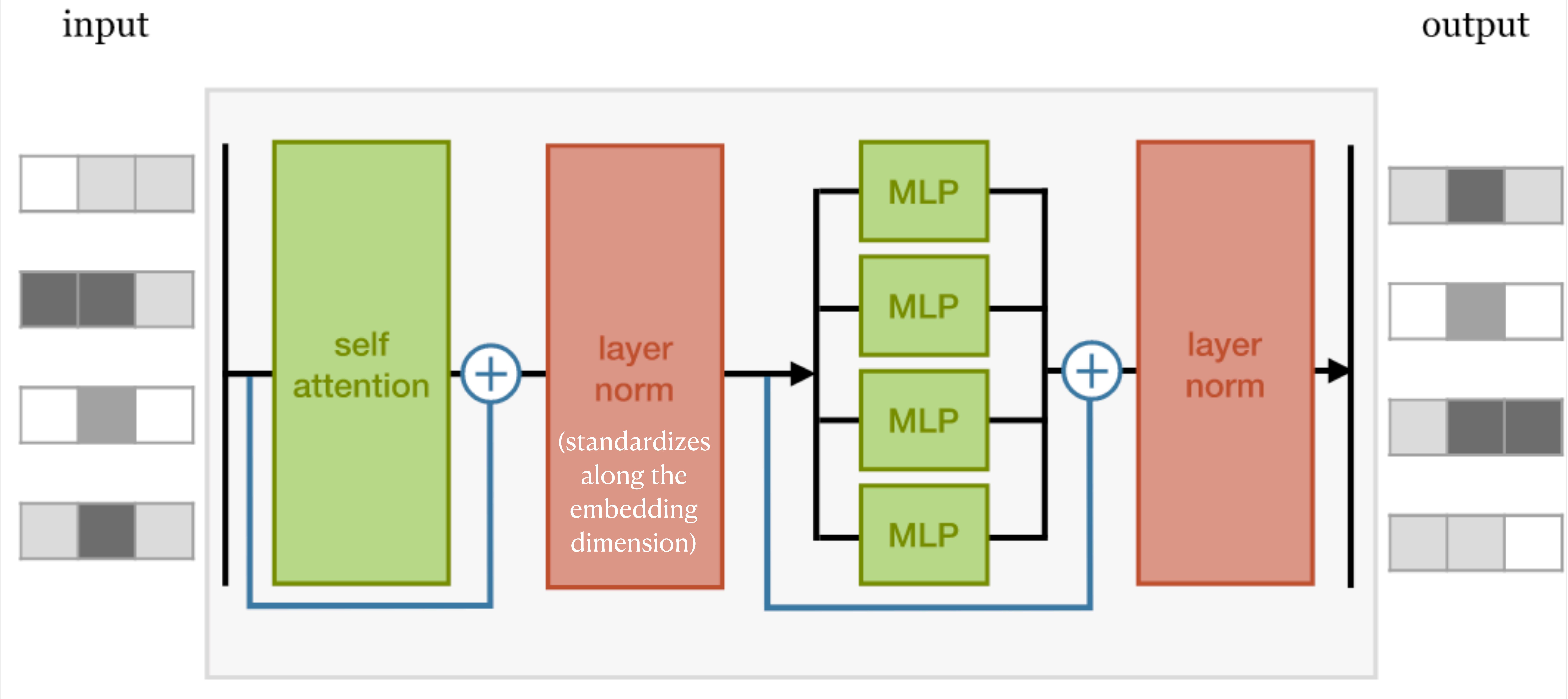
1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Masked self-attention

Causal transformer, 1706.03762



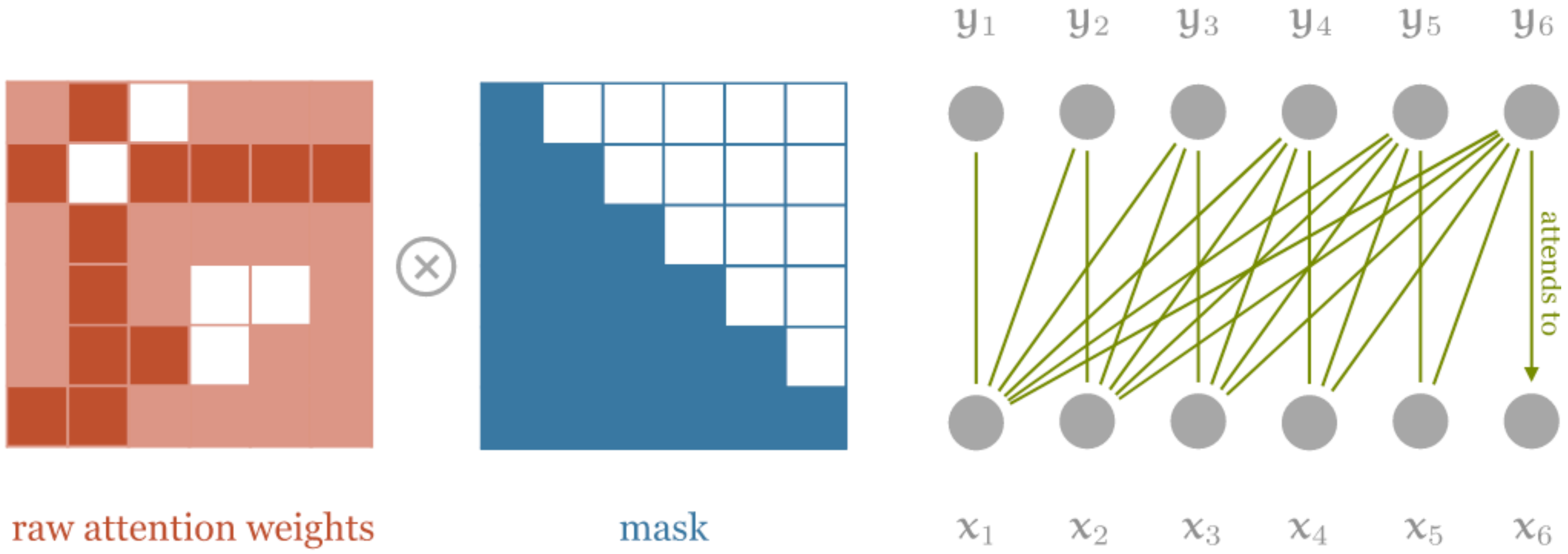
The transformer block



https://peterbloem.nl/blog/transformers

Masked self-attention

$$y_i = \sum_j \underbrace{\alpha(x_i, x_j)}_{\text{attention weight}} x_j$$



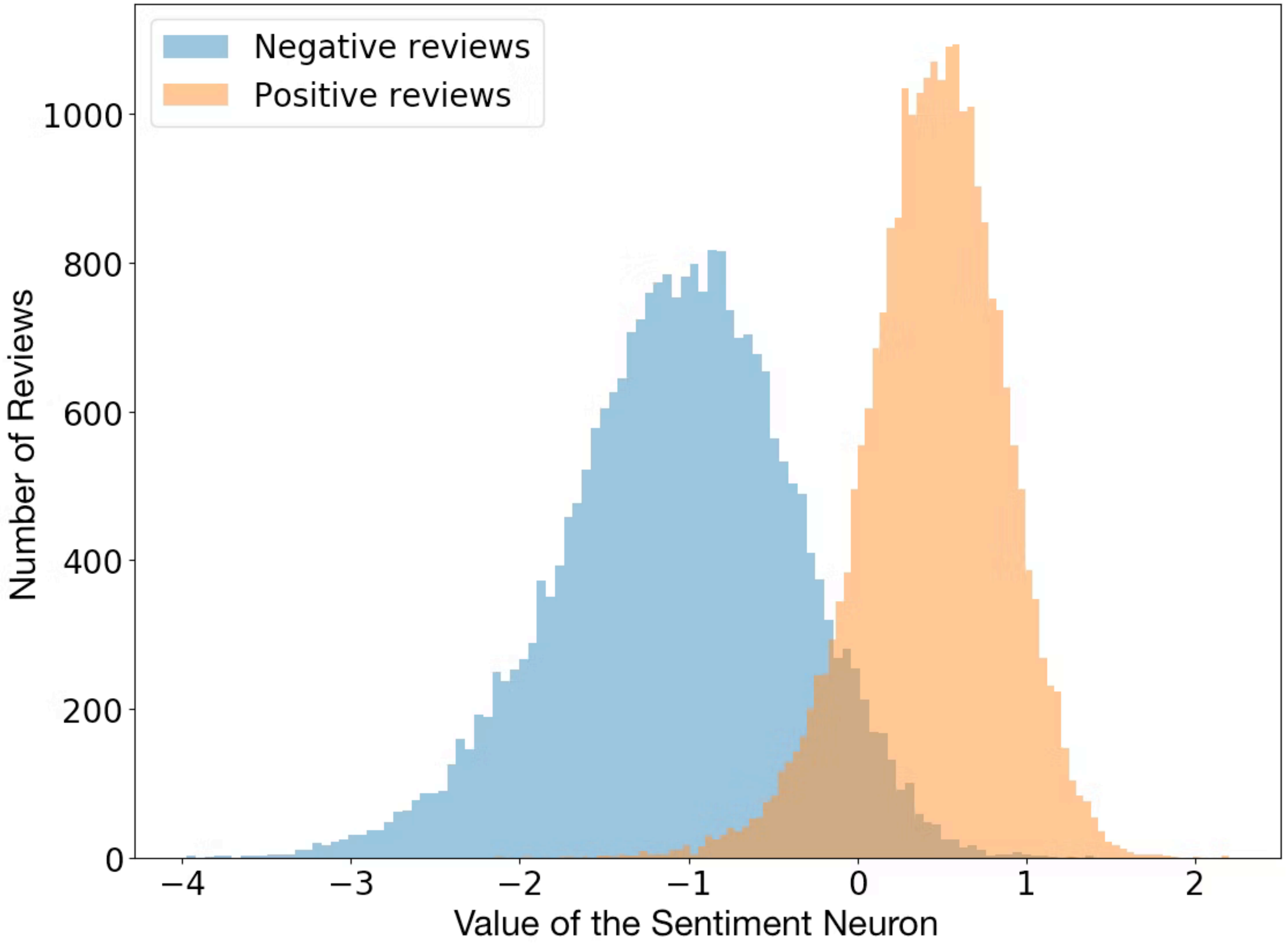
Learning to Generate Reviews and Discovering Sentiment

Alec Radford [1] **Rafal Jozefowicz** [1] **Ilya Sutskever** [1]

1704.01444

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 [1] Alec Radford [1] Rewon Child [1] Jeff Wu [1] Heewoo Jun [1] Prafulla Dhariwal [1] David Luan [1]
Ilya Sutskever [1]

Representation learned by image GPT

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.

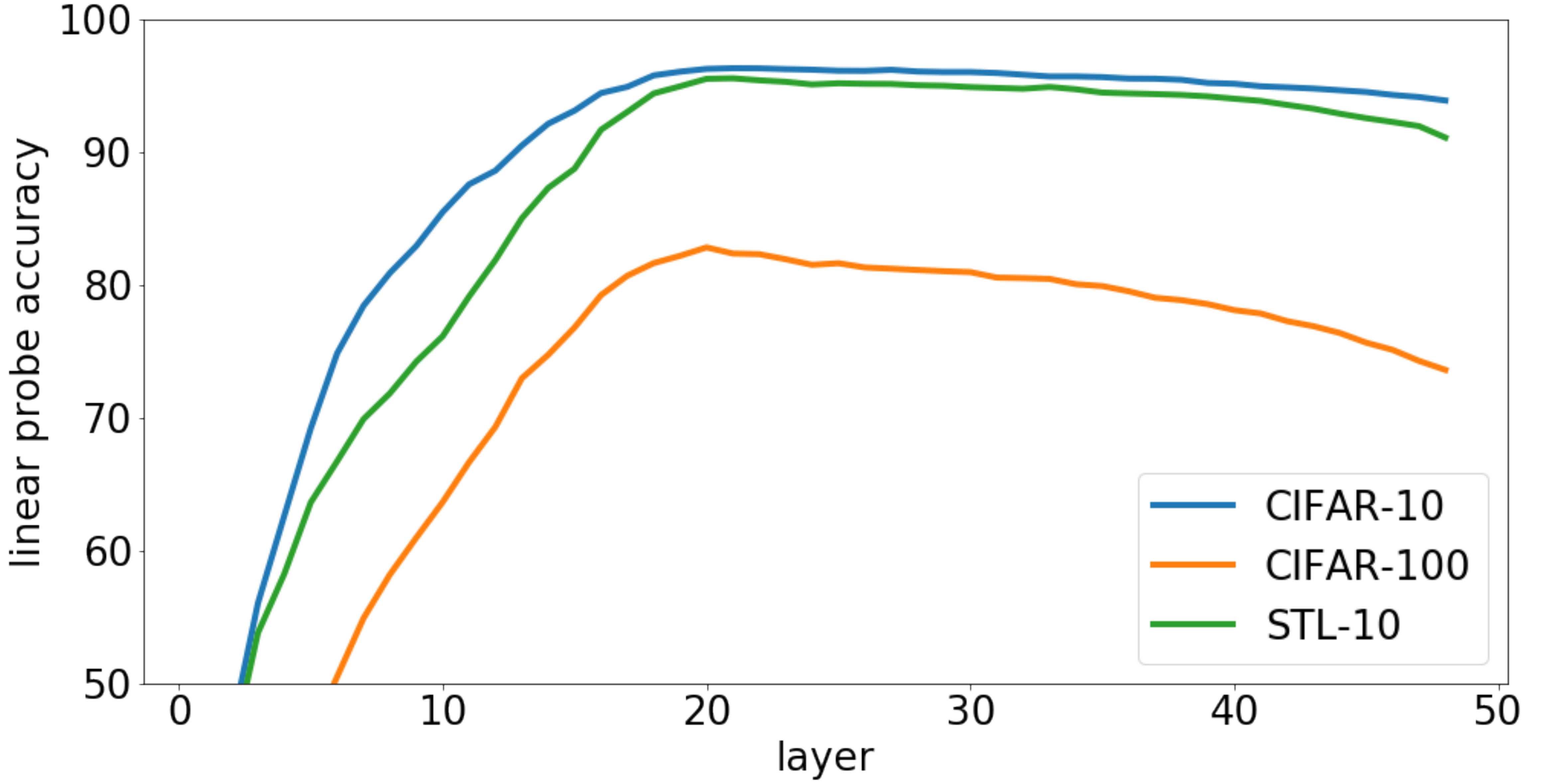
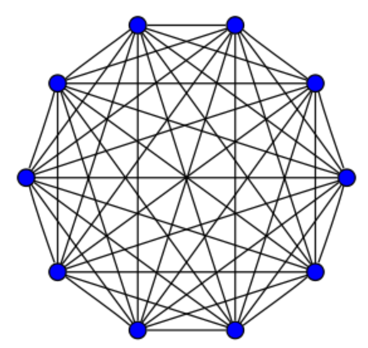
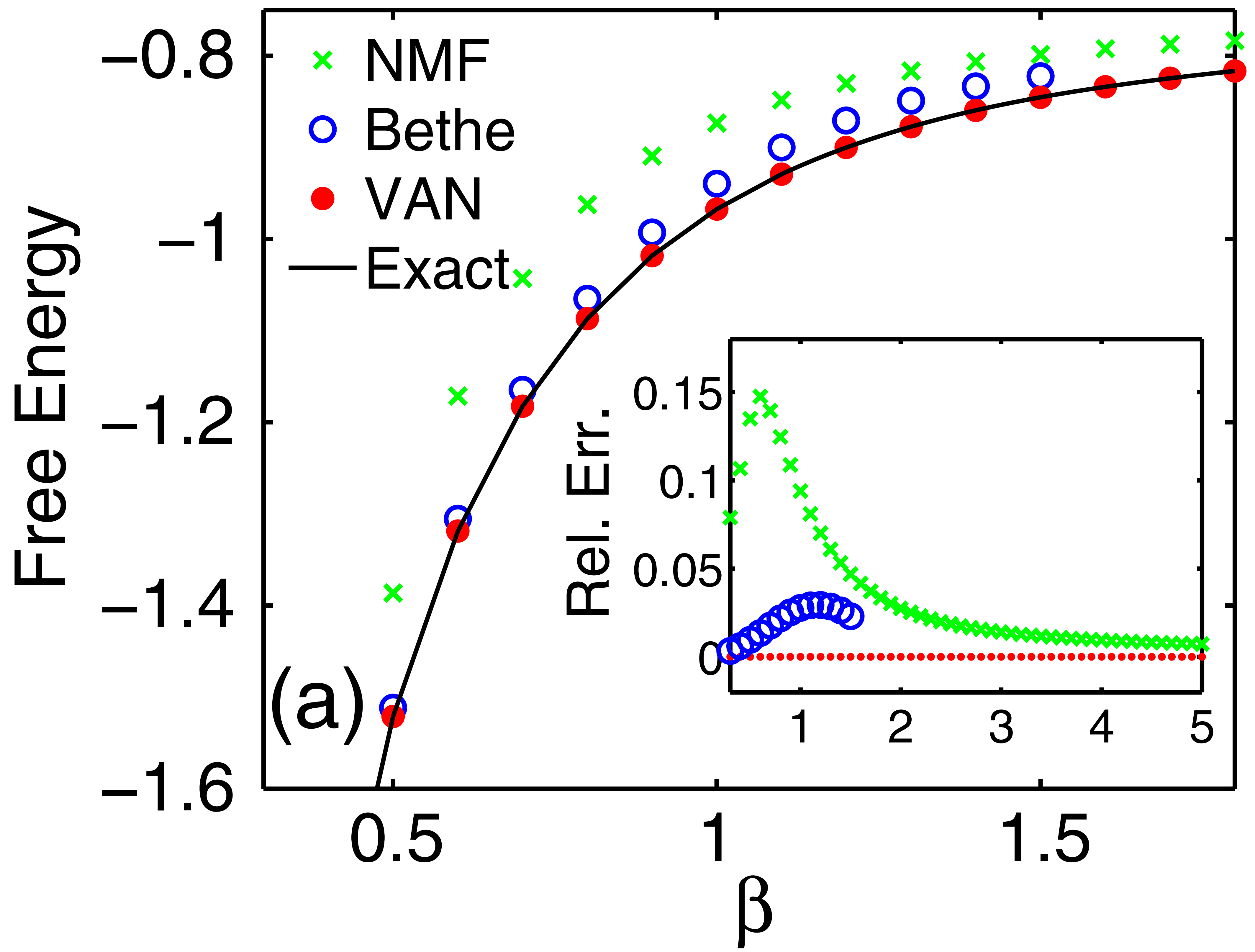


Figure 2. Representation quality depends on the layer from which we extract features. In contrast with supervised models, the best representations for these generative models lie in the middle of the network. We plot this unimodal dependence on depth by showing linear probes for iGPT-L on CIFAR-10, CIFAR-100,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

Variational autoregressive network for statistical mechanics



Sherrington-Kirkpatrick spin glass



Naive mean-field
factorized probability

$$p(\boldsymbol{X}) = \prod_i p(x_i)$$

Bethe approximation
pairwise interaction

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

Variational autoregressive
network

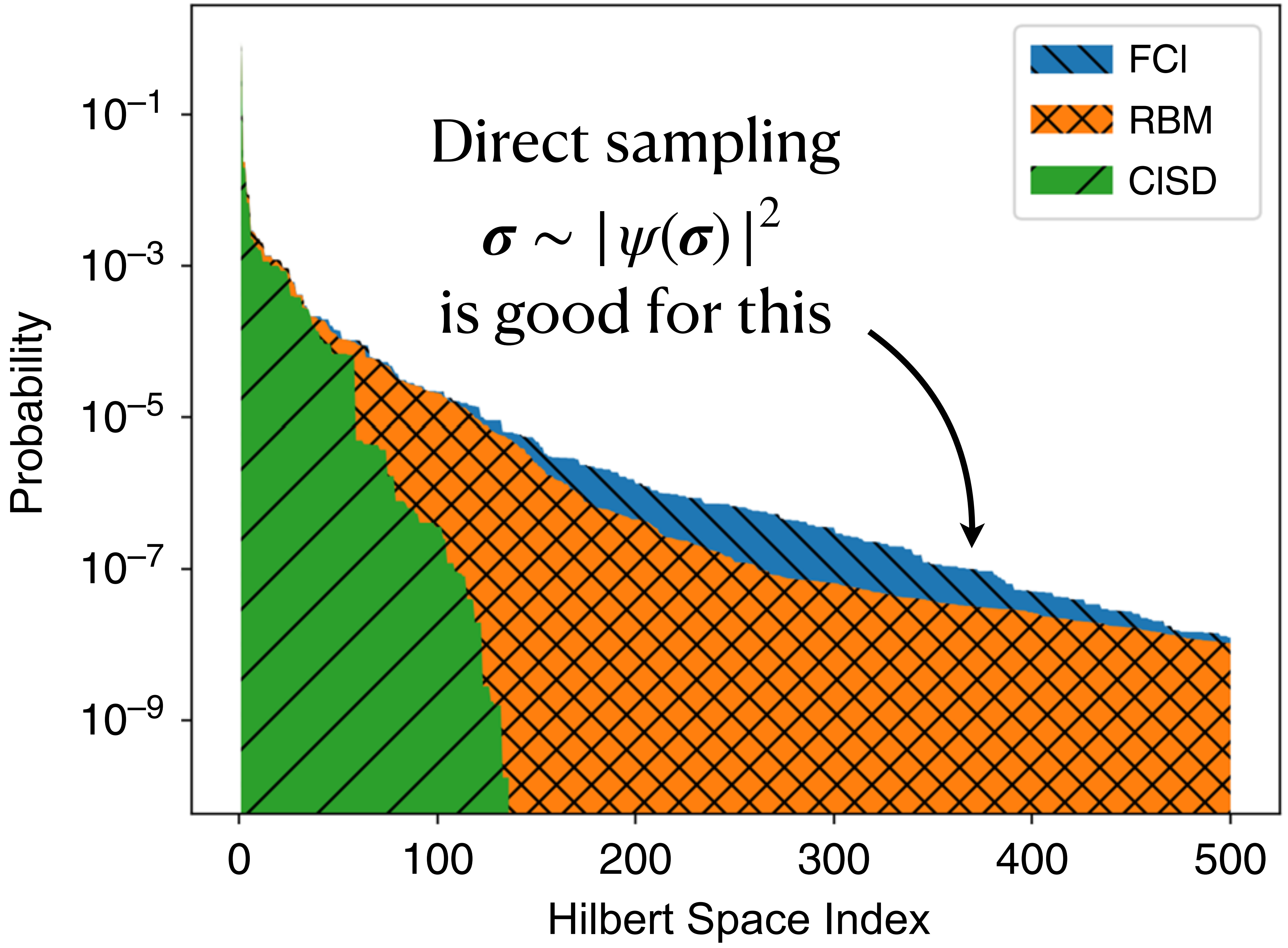
$$p(\boldsymbol{X}) = \prod_i p(x_i | \boldsymbol{x}_{<i})$$

Wu, LW, Zhang, PRL '19

github.com/wdphy16/stat-mech-van

Variational autoregressive quantum states

$$\psi(\boldsymbol{\sigma}) = \psi(\sigma_1)\psi(\sigma_2\,|\,\sigma_1)\psi(\sigma_3\,|\,\sigma_1,\sigma_2)\cdots$$



N2 molecule, Choo et al, Nat. Comm. '20

Objective function: ground state energy

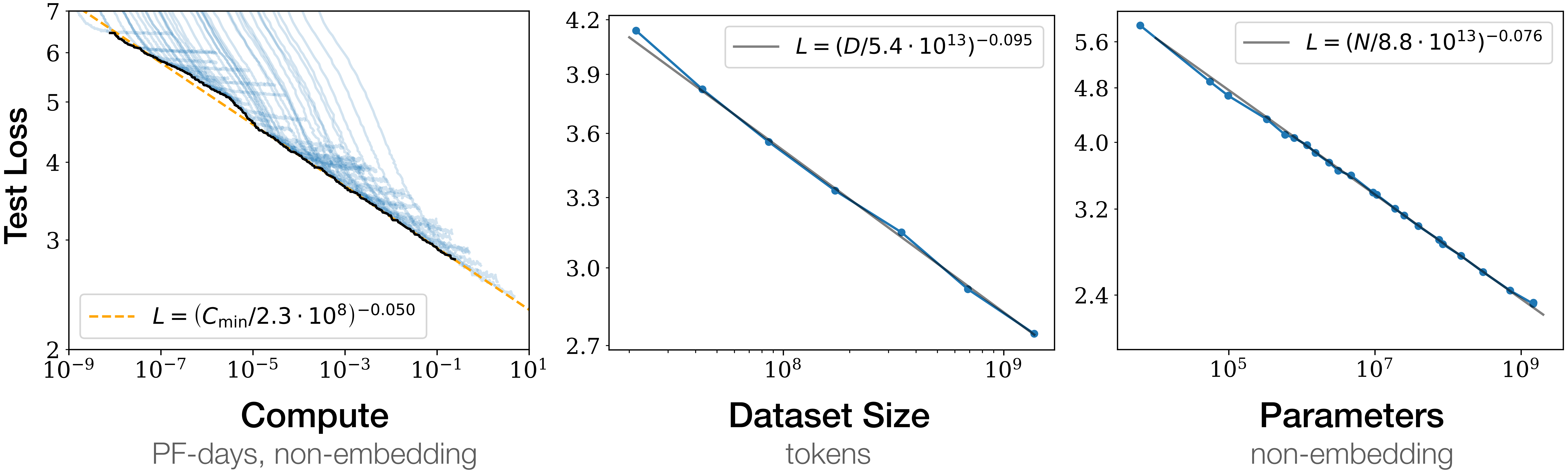
McMillan 1965, Carleo & Troyer Science 2017

$$\frac{\langle\psi\,|\,\hat{H}\,|\,\psi\rangle}{\langle\psi\,|\,\psi\rangle} = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}\sim|\psi(\boldsymbol{\sigma})|^2}\left[\frac{\hat{H}\psi(\boldsymbol{\sigma})}{\psi(\boldsymbol{\sigma})}\right]$$

Sharir, Levine, Wies, Carleo, Shashua, PRL '20
Hibat-Allah, Ganahl, Hayward, Melko, Carrasquilla, PRResarch '20
Barrett et al, Nat. Mach. Intell. '22
Zhao et al, MLST. '23
Shang et al, 2307.09343

Scaling law

Kaplan et al, 2001.08361

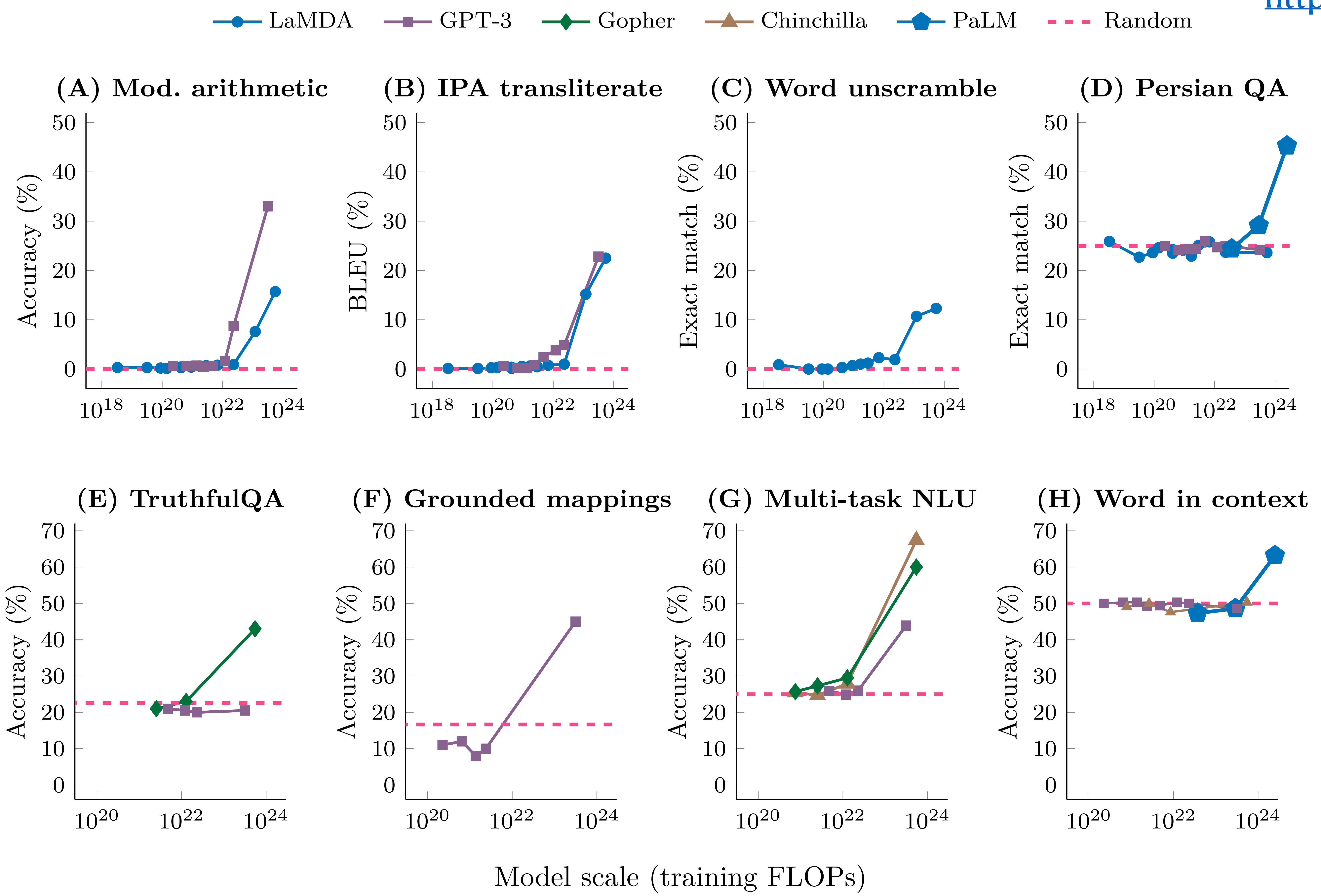


"It would also be exciting to find a theoretical framework from which the scaling relations can be derived: a 'statistical mechanics' underlying the 'thermodynamics' we have observed."

Emergent abilities: more is different

Wei et al, 2206.07682

https://www.jasonwei.net/blog/emergence



2304.15004

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo

Computer Science, Stanford University

$$\mathrm{Accuracy}(N) \approx p_N(\text{single token correct})^{\text{num. of tokens}} = \exp\Big(-(N/c)^{\alpha}\Big)^{L}$$

"The researcher's choice of metric can nonlinearly and/or discontinuously transform the error rate in a manner that causes the model performance to appear sharp and unpredictable.“

Generative models and their physics genes

Goodfellow,
NIPS tutorial, 1701.00160

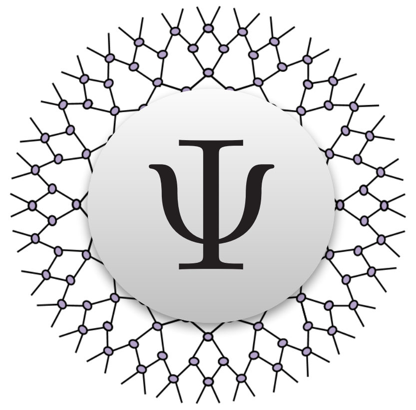
$p(\boldsymbol{x})$

Explicit density

Implicit density

Direct

GAN



Tensor Networks

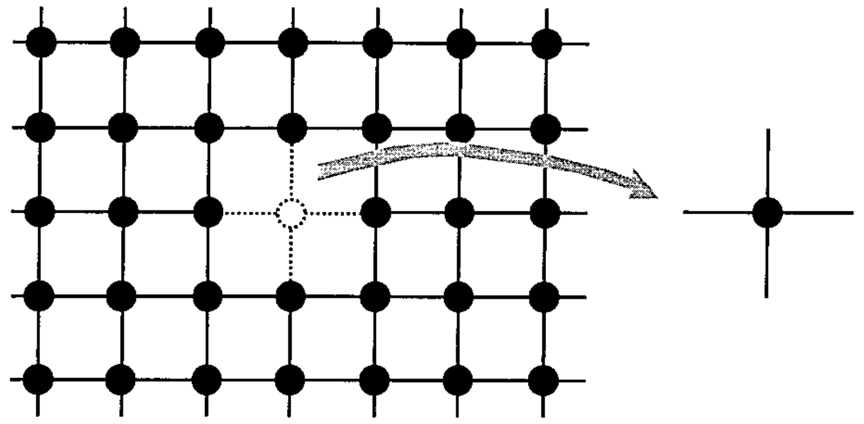
Tractable density

-Fully visible belief nets
 -NADE
 -MADE
 -PixelRNN
-Change of variables models (nonlinear ICA)

Approximate density

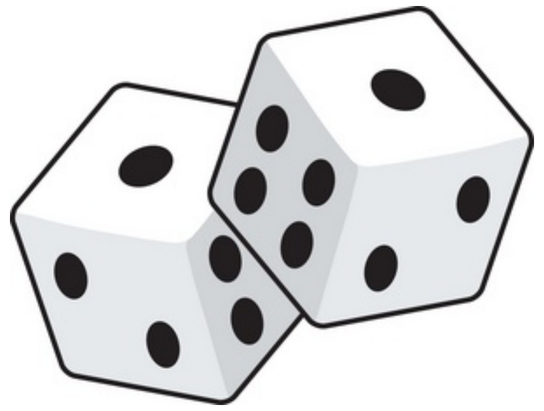
Variational

Variational autoencoder



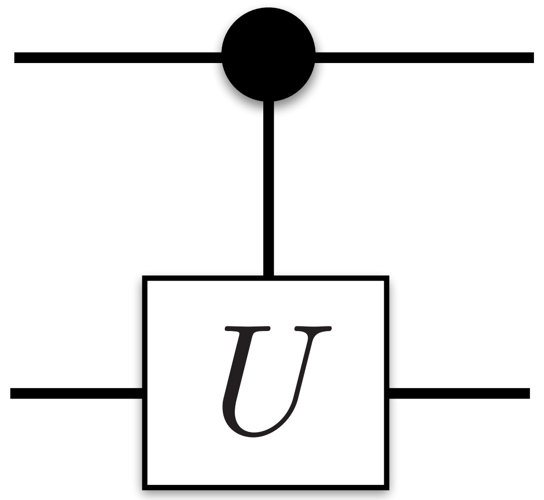
Markov Chain

Boltzmann machine



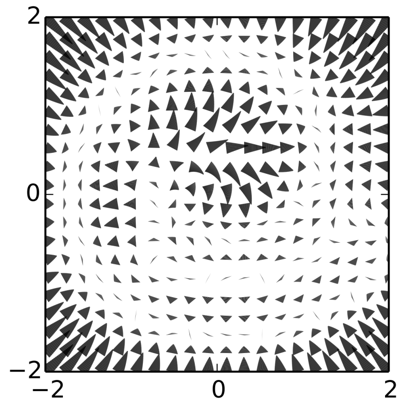
Markov Chain

GSN



Quantum Circuits

+**Diffusion models**



Normalizing flows

Parallel WaveNet 1711.10433

https://deepmind.com/blog/high-fidelity-speech-synthesis-wavenet/

Glow 1807.03039

https://blog.openai.com/glow/

Normalizing flows

Parallel WaveNet 1711.10433

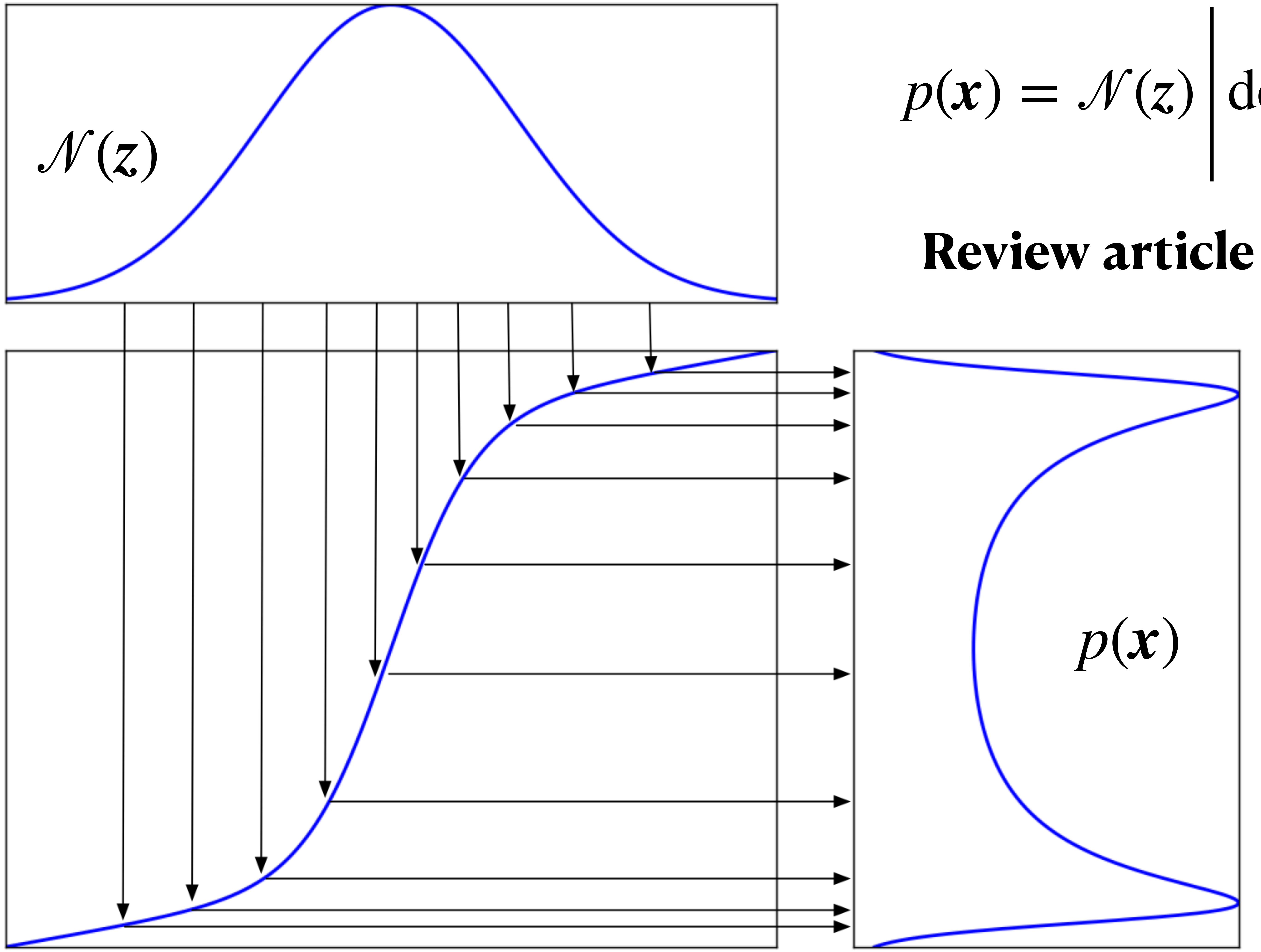
https://deepmind.com/blog/high-fidelity-speech-synthesis-wavenet/

Glow 1807.03039

https://blog.openai.com/glow/

Normalizing flow in a nutshell

Base density



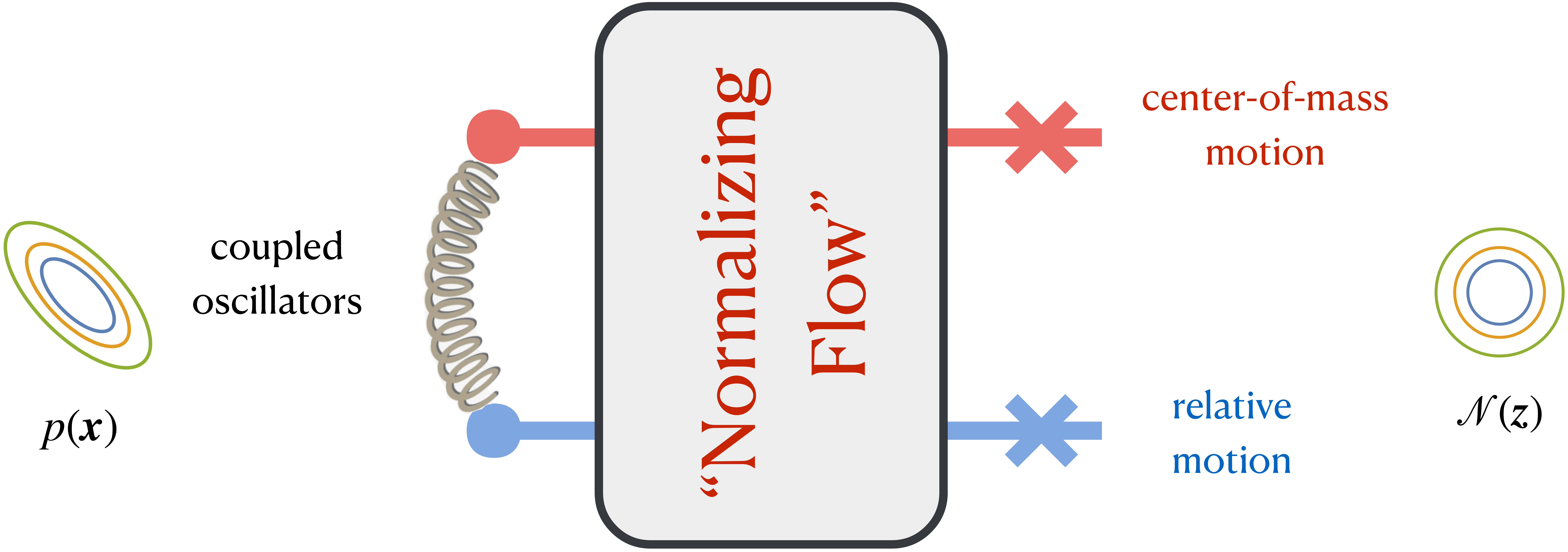
$$p(\boldsymbol{x}) = \mathcal{N}(z)\left|\det\left(\frac{\partial z}{\partial \boldsymbol{x}}\right)\right|$$

Review article 1912.02762

“neural net” with 1 neuron

Target density

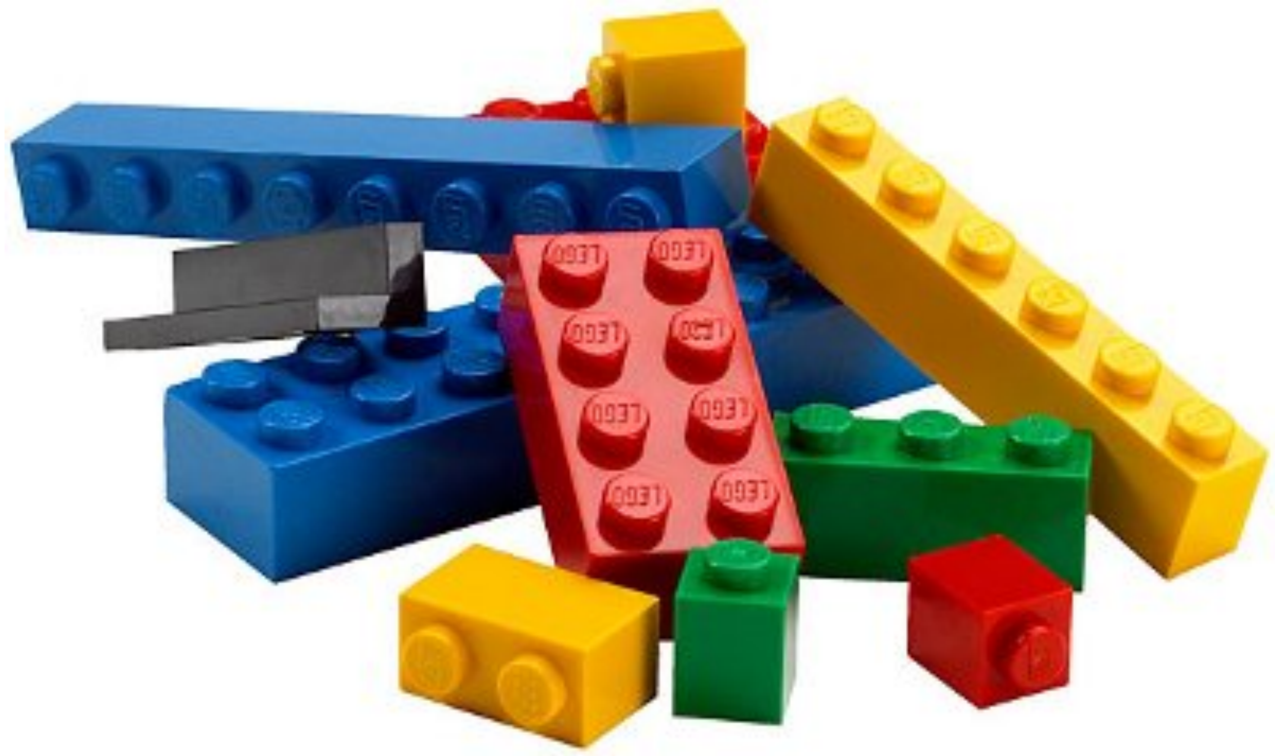
Physics intuition of normalizing flow



High-dimensional, nonlinear, learnable, composable transformations

Flow architecture design

Composability

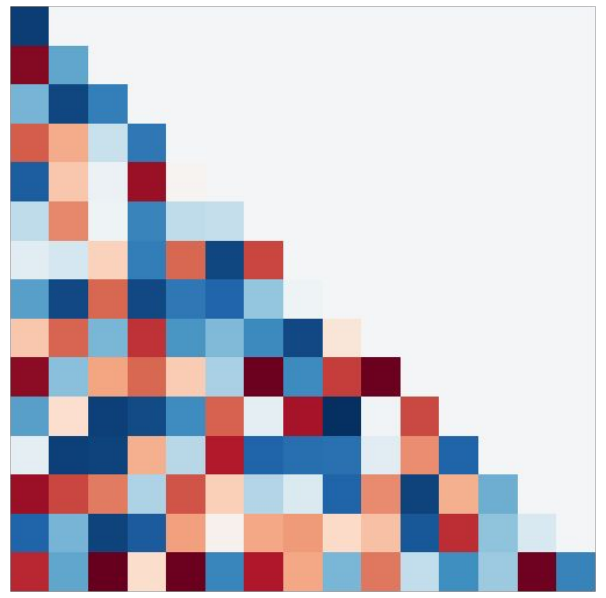


$$z = \mathcal{T}(x)$$

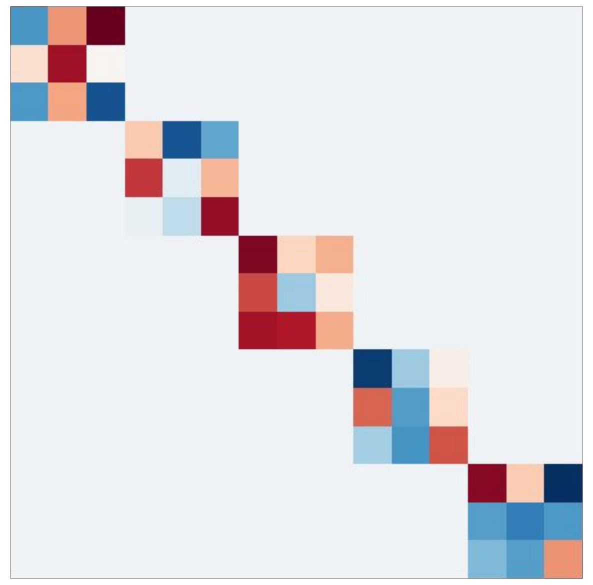
$$\mathcal{T} = \mathcal{T}_1 \circ \mathcal{T}_2 \circ \mathcal{T}_3 \circ \cdots$$

Balanced efficiency & inductive bias

$$\left| \det \left(\frac{\partial z}{\partial x} \right) \right|$$



Autoregressive



Blockwise

$$\frac{\partial p(x, t)}{\partial t} + \nabla \cdot \left[p(x, t) v \right] = 0$$

Continuous flow

Example of a building block

Forward

arbitrary neural nets

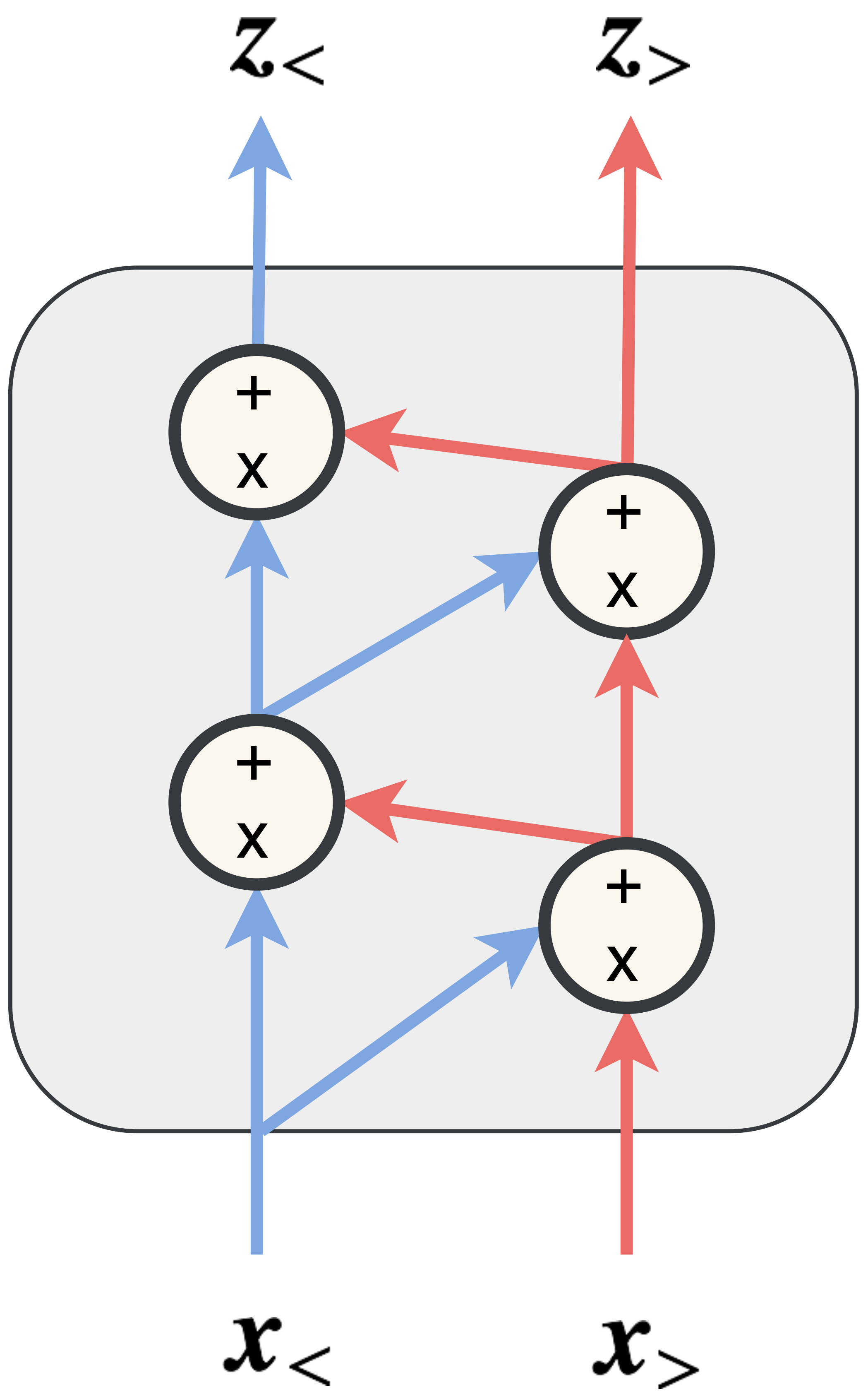
$$\begin{cases} \boldsymbol{x}_< = z_< \\ \boldsymbol{x}_> = z_> \odot e^{s(z_<)} + t(z_<) \end{cases}$$

Inverse

$$\begin{cases} z_< = \boldsymbol{x}_< \\ z_> = (\boldsymbol{x}_> - t(\boldsymbol{x}_<)) \odot e^{-s(\boldsymbol{x}_<)} \end{cases}$$

Log-Abs-Jacobian-Det

$$\ln \left| \det \left(\frac{\partial x}{\partial z} \right) \right| = \sum_i [s(z_<)]_i$$

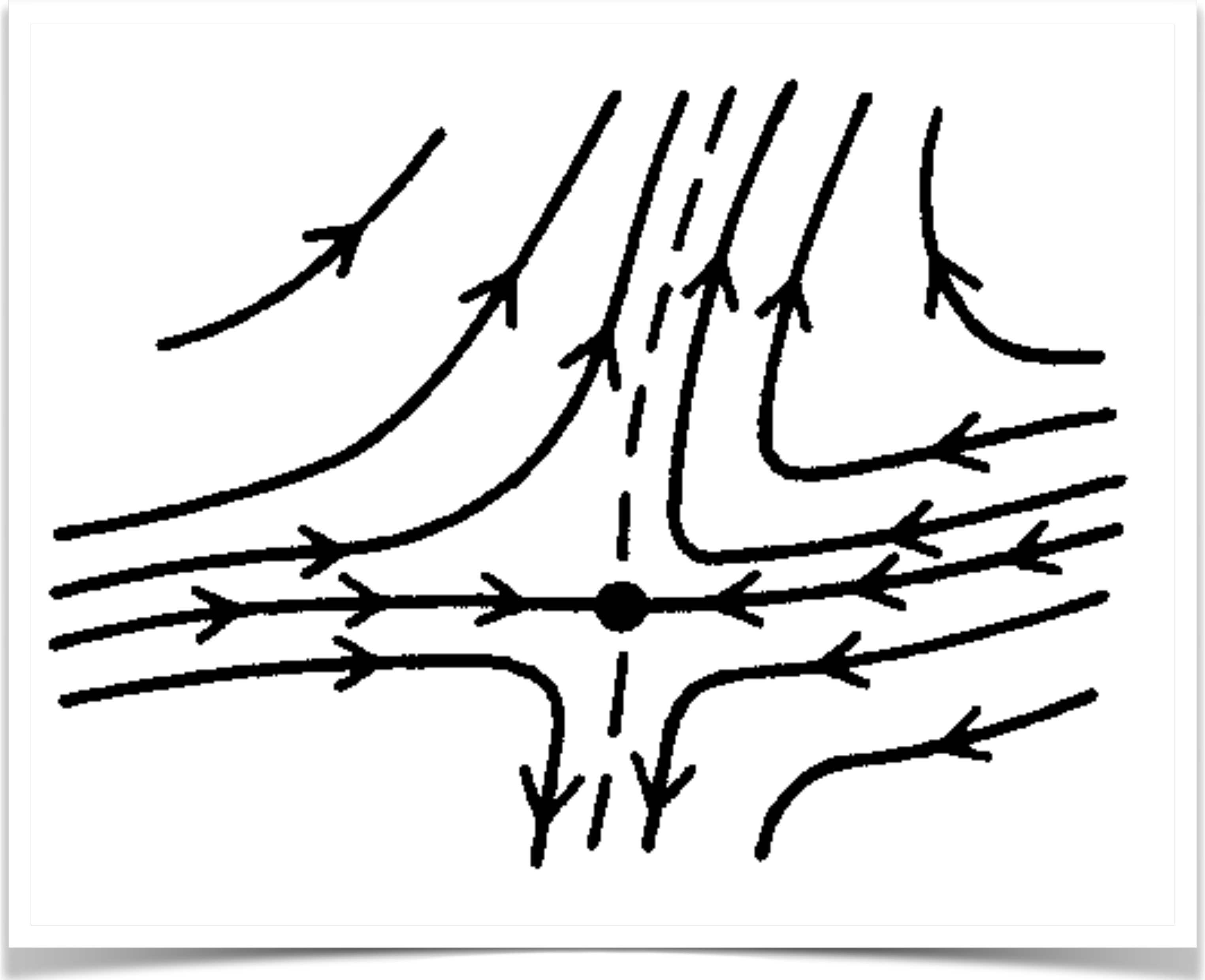


Real NVP, Dinh et al,1605.08803

Turns out to have surprising connection Störmer–Verlet integration

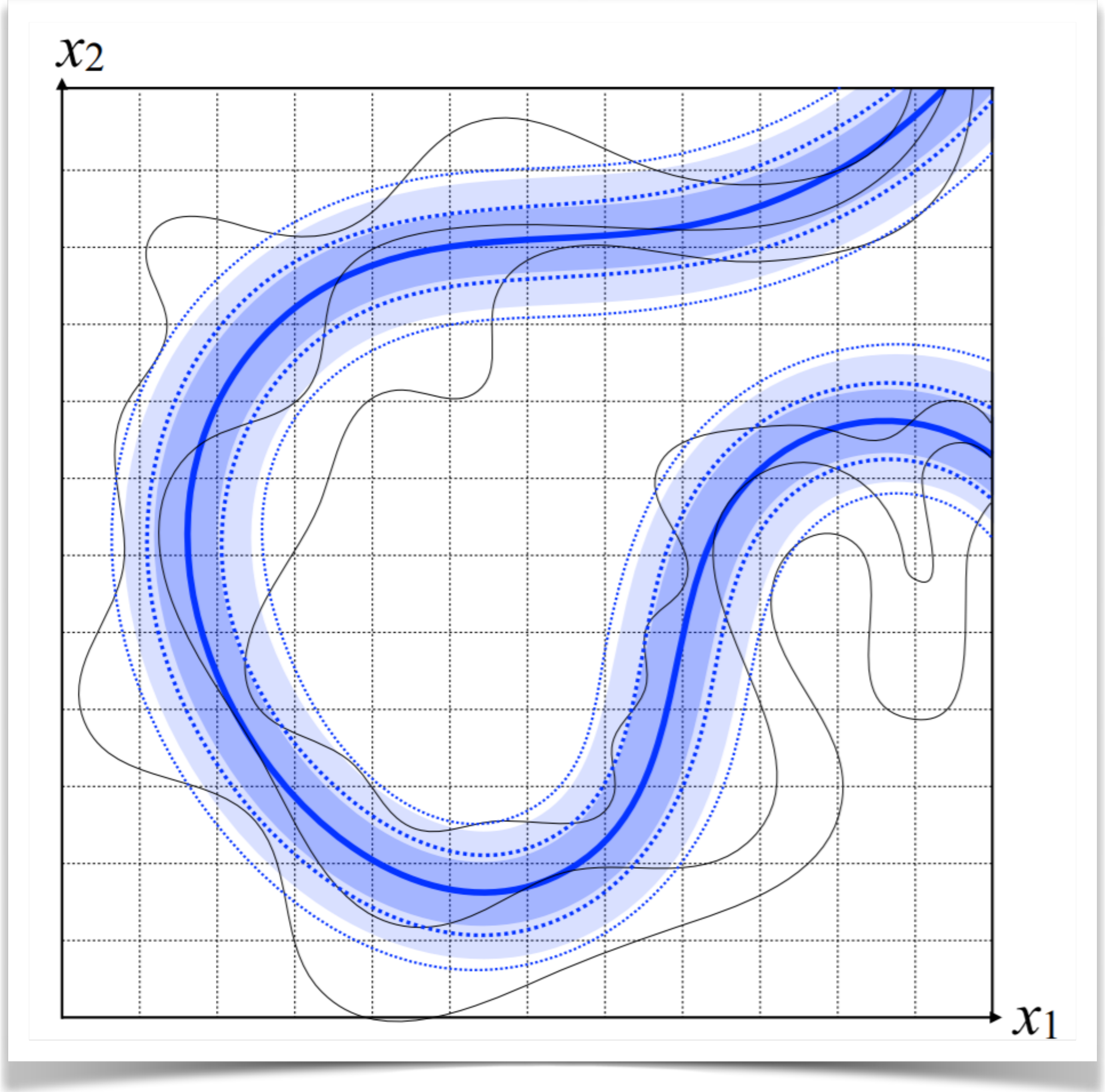
Why is flow useful for physics?

Renormalization group



Effective theory emerges upon transformation of the variables

Monte Carlo update

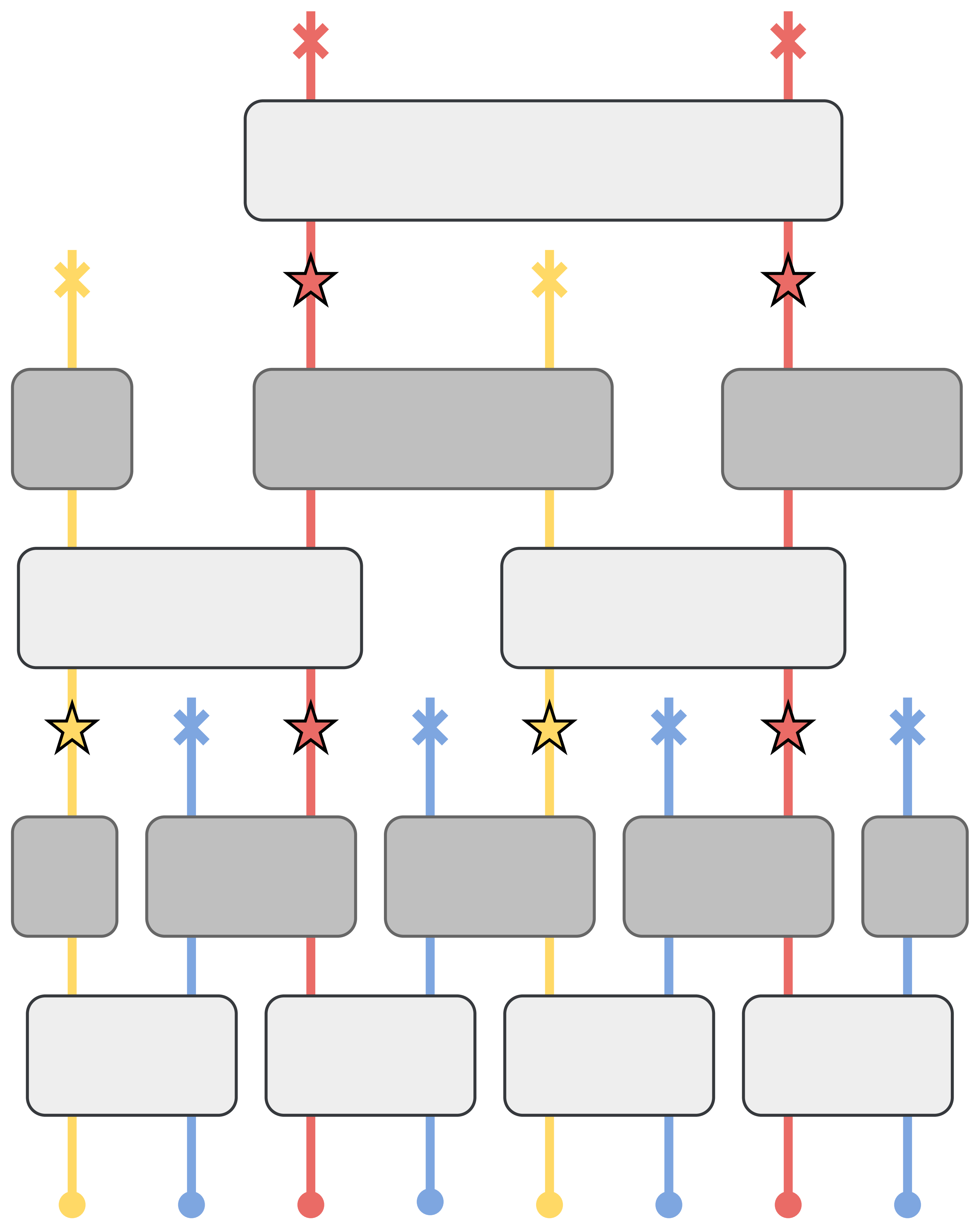


Physics happens on a manifold
Train neural nets to unfold that manifold

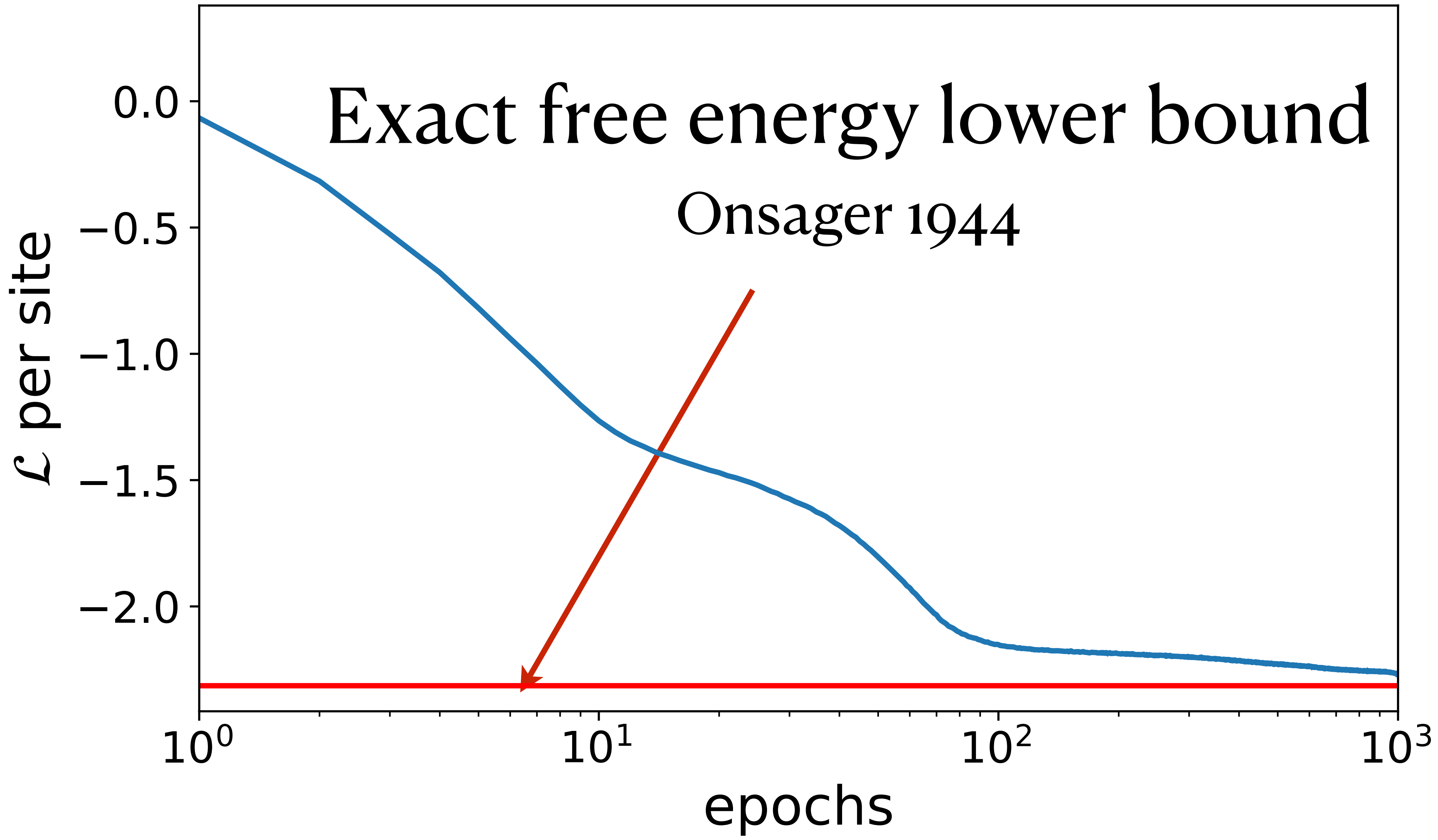
Neural network renormalization group

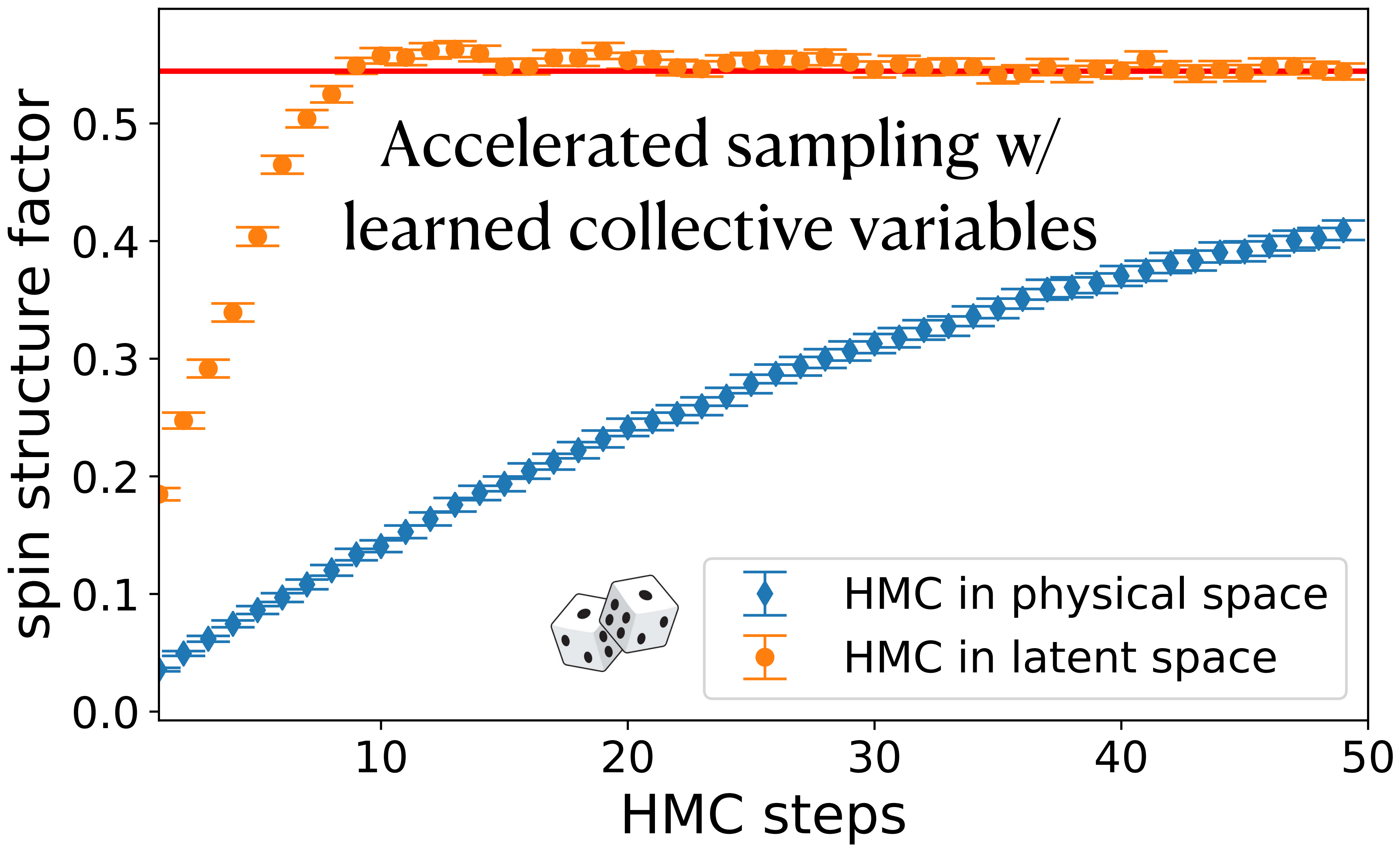
Li, LW, PRL '18 [li012589/NeuralRG](https://github.com/li012589/NeuralRG)

Collective variables



Physical variables

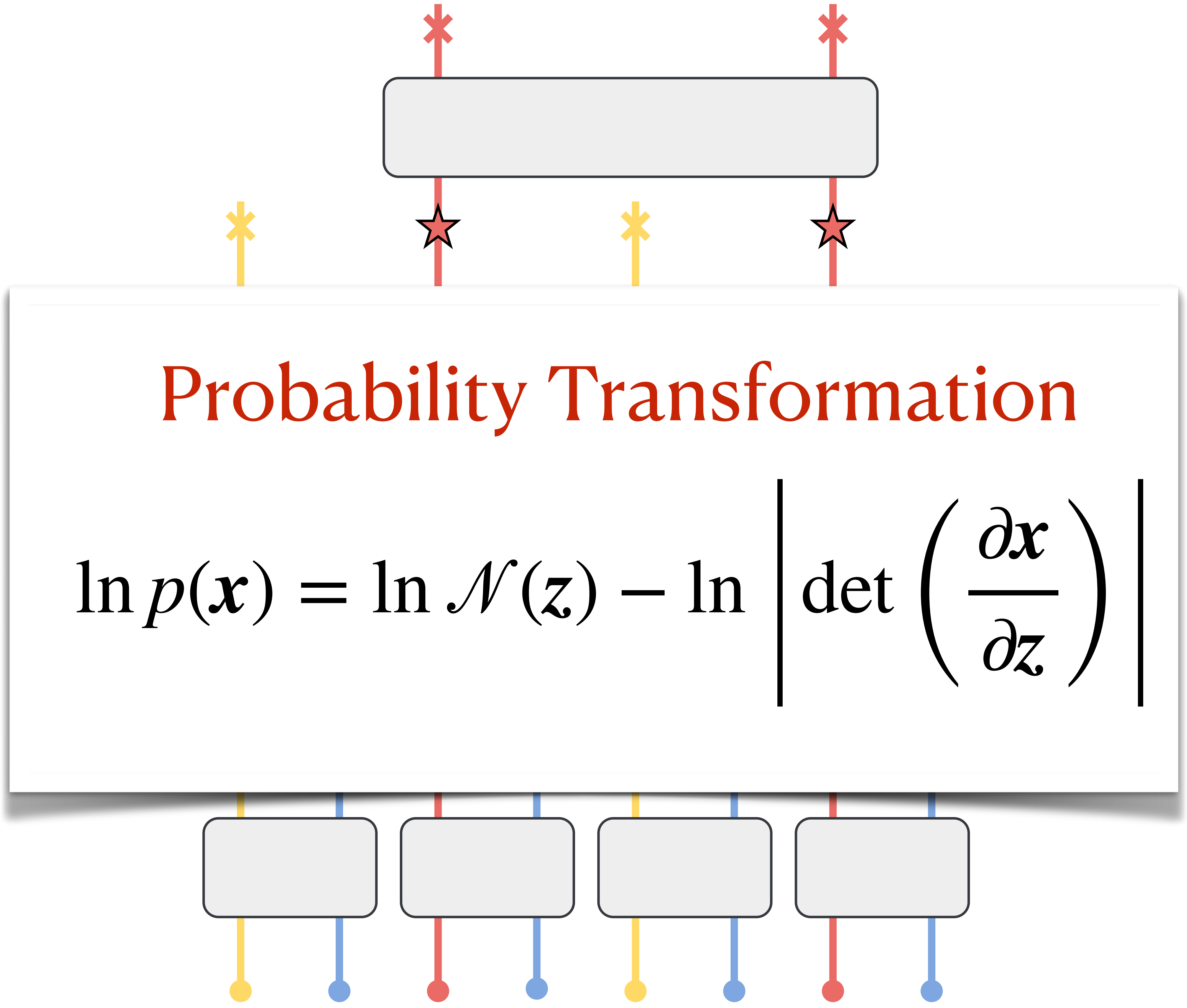




Neural network renormalization group

Li, LW, PRL '18 [li012589/NeuralRG](https://github.com/li012589/NeuralRG)

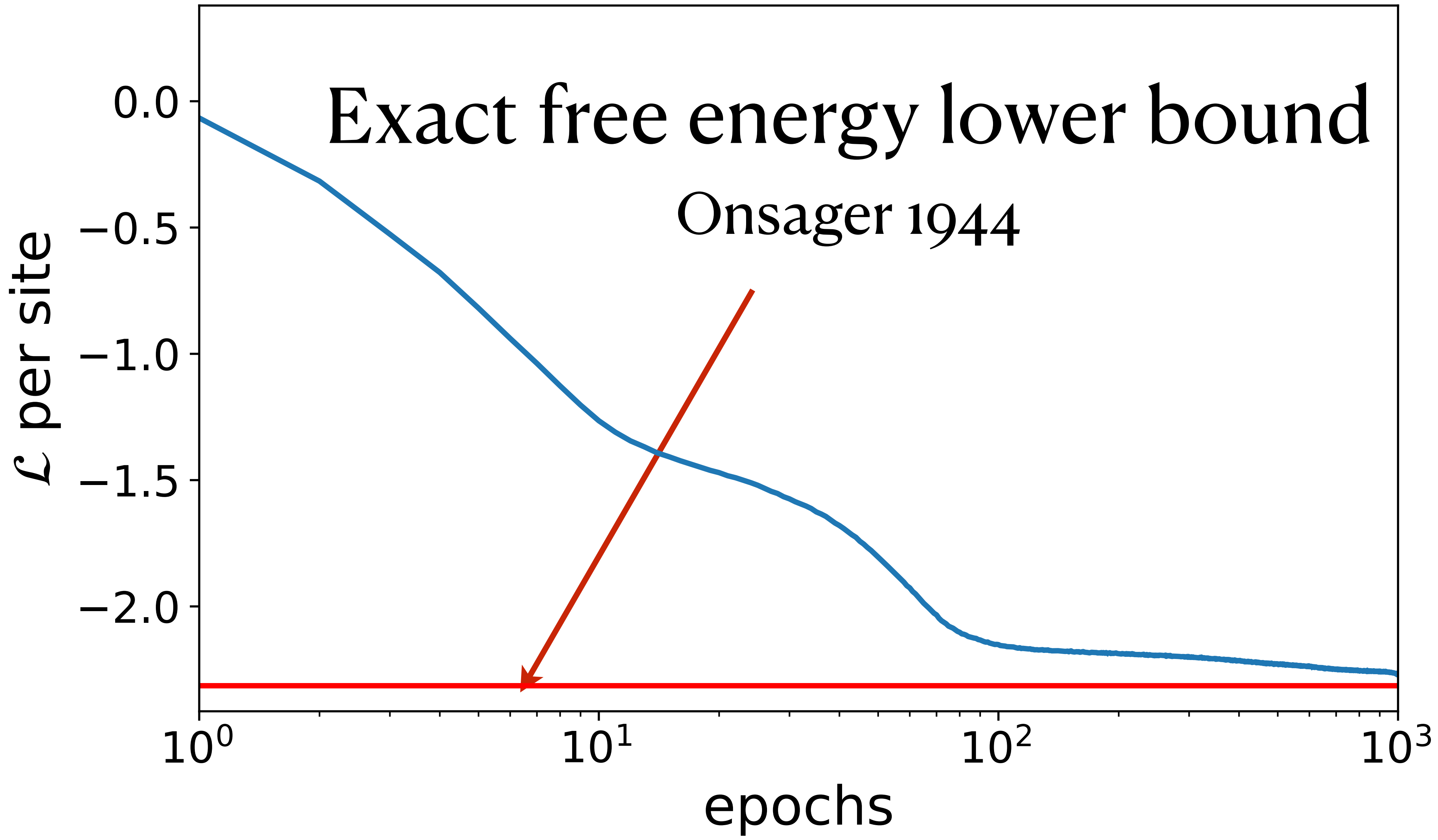
Collective variables

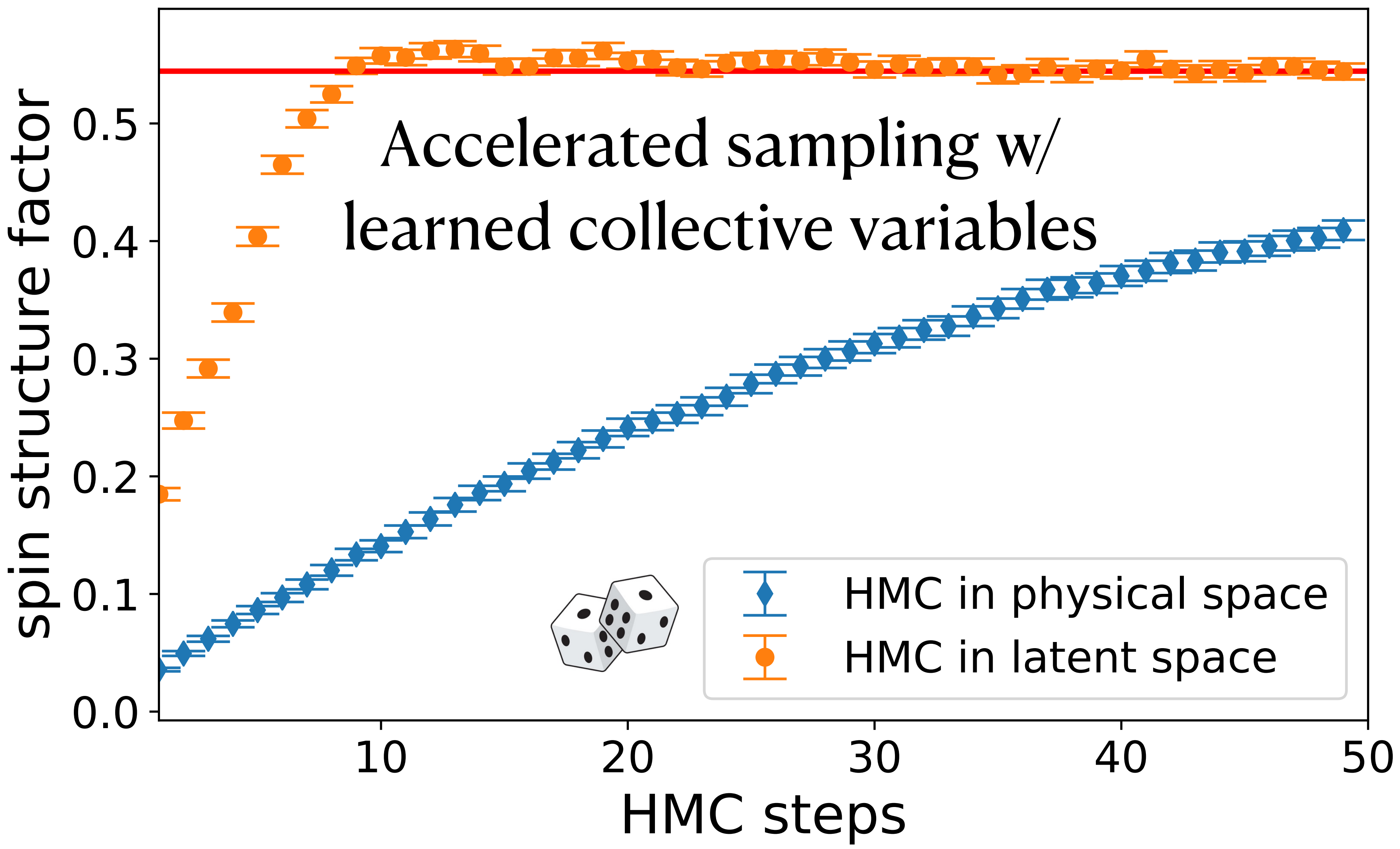


Probability Transformation

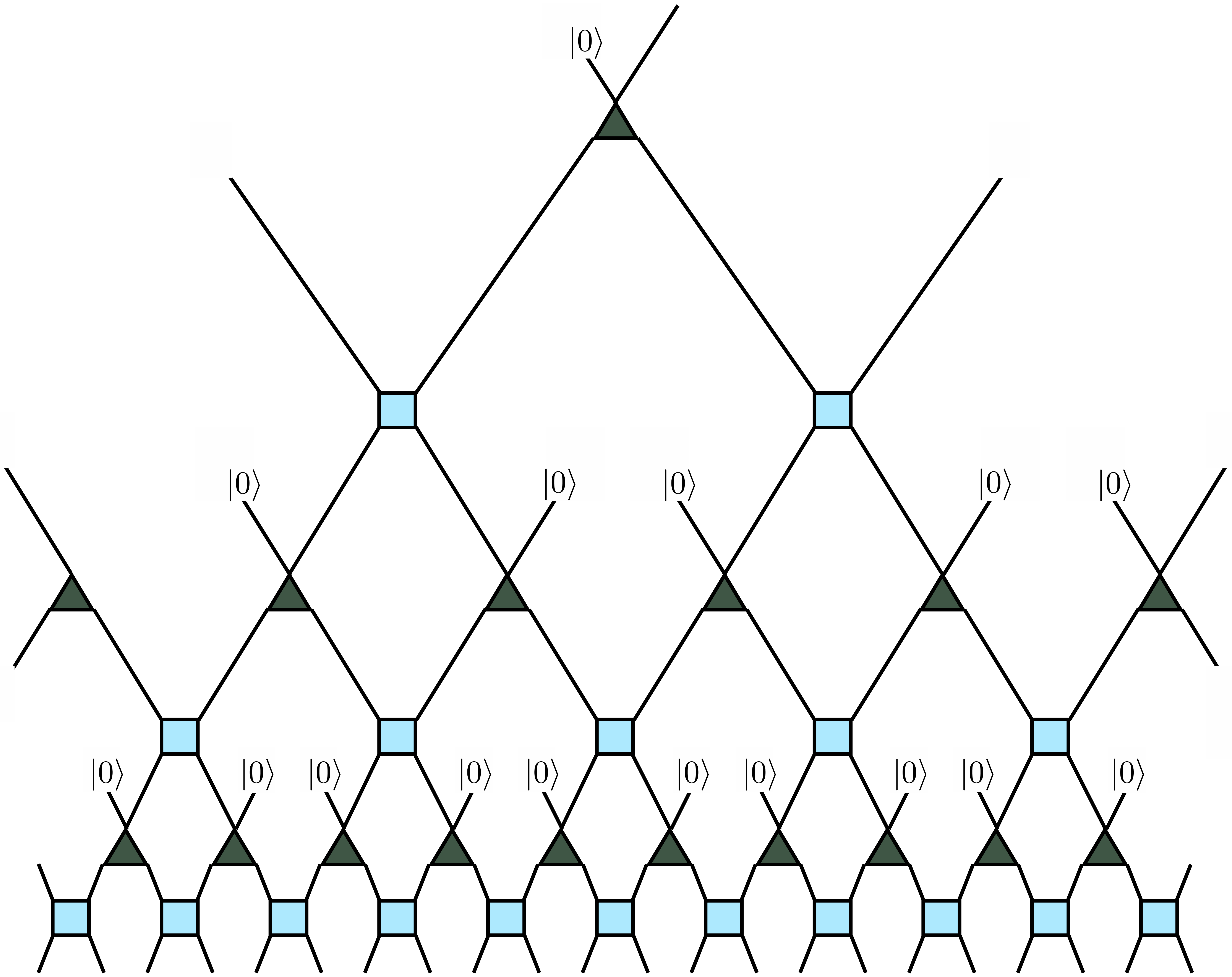
$$\ln p(\boldsymbol{x}) = \ln \mathcal{N}(\boldsymbol{z}) - \ln \left| \det \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} \right) \right|$$

Physical variables

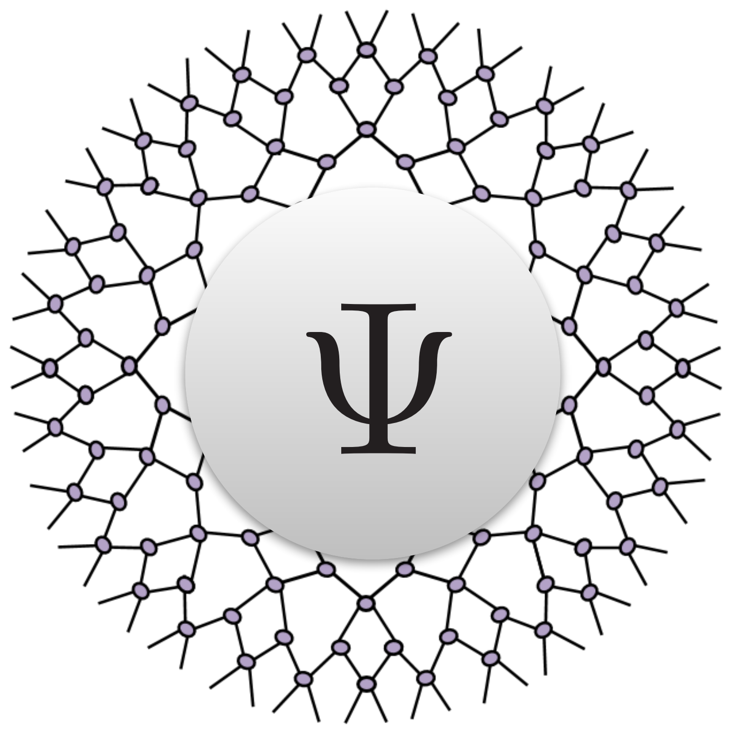




Quantum version of the architecture

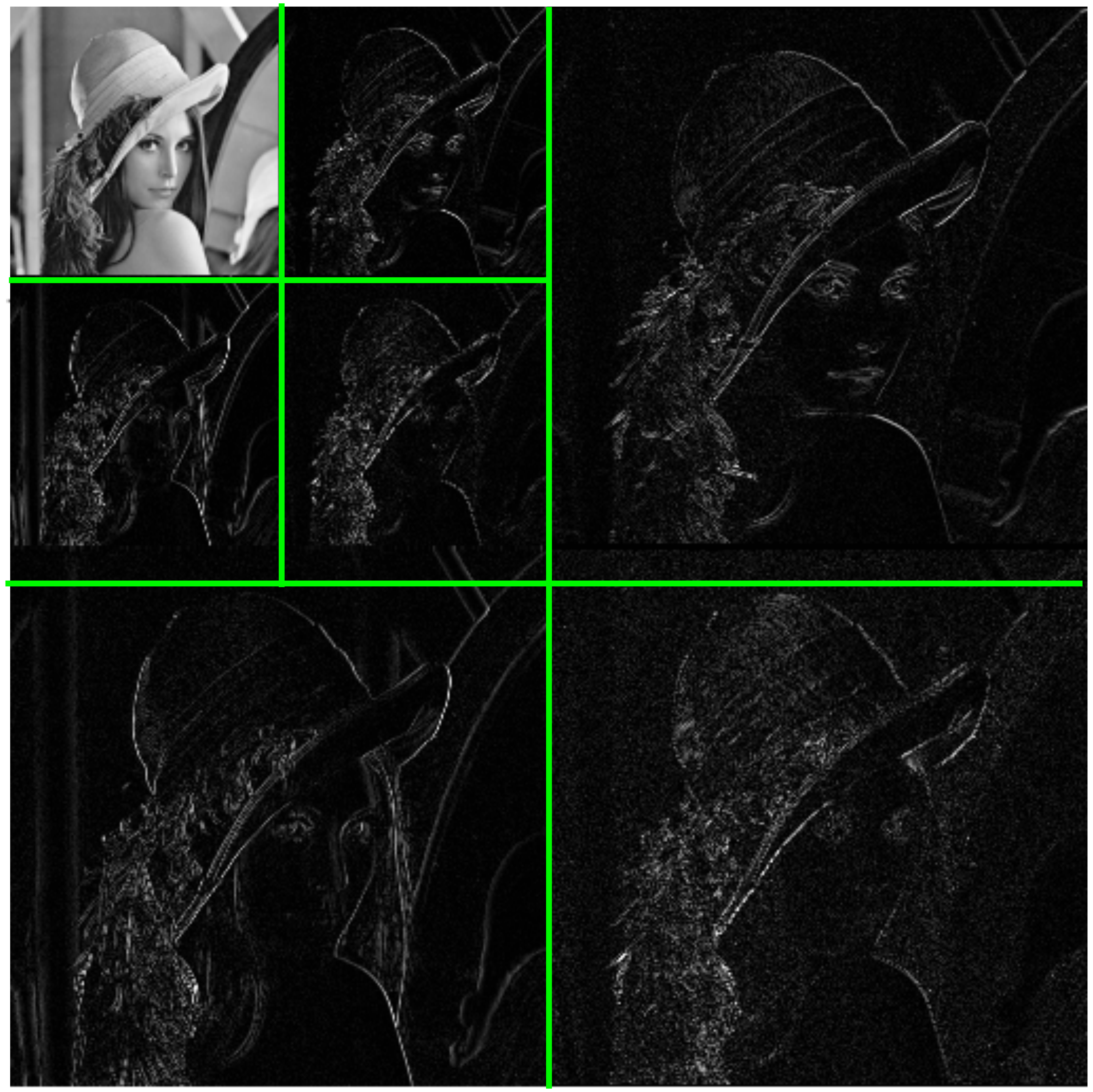


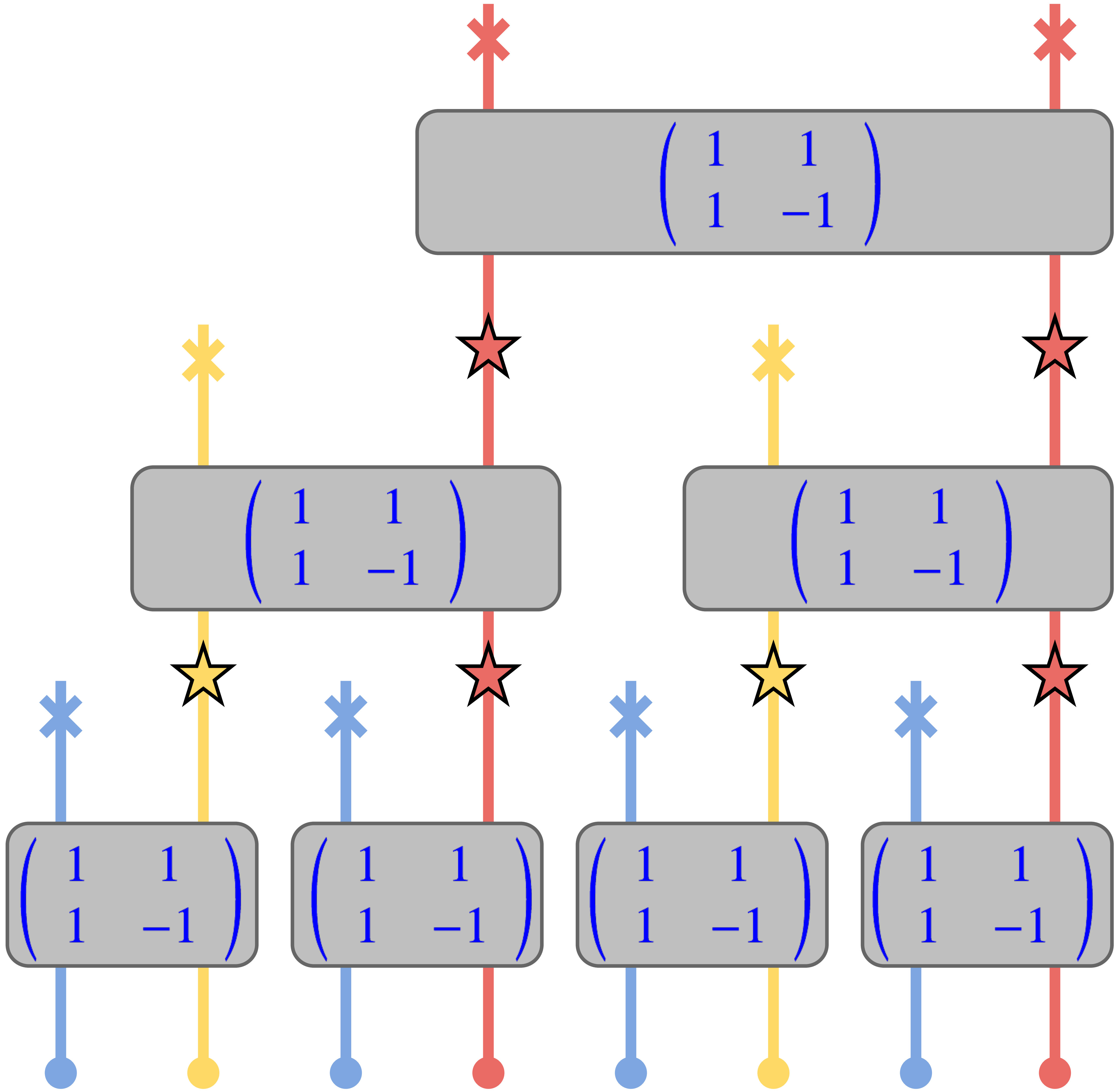
Entangled qubits



Multi-Scale
Entanglement
Renormalization
Ansatz

Connection to wavelets

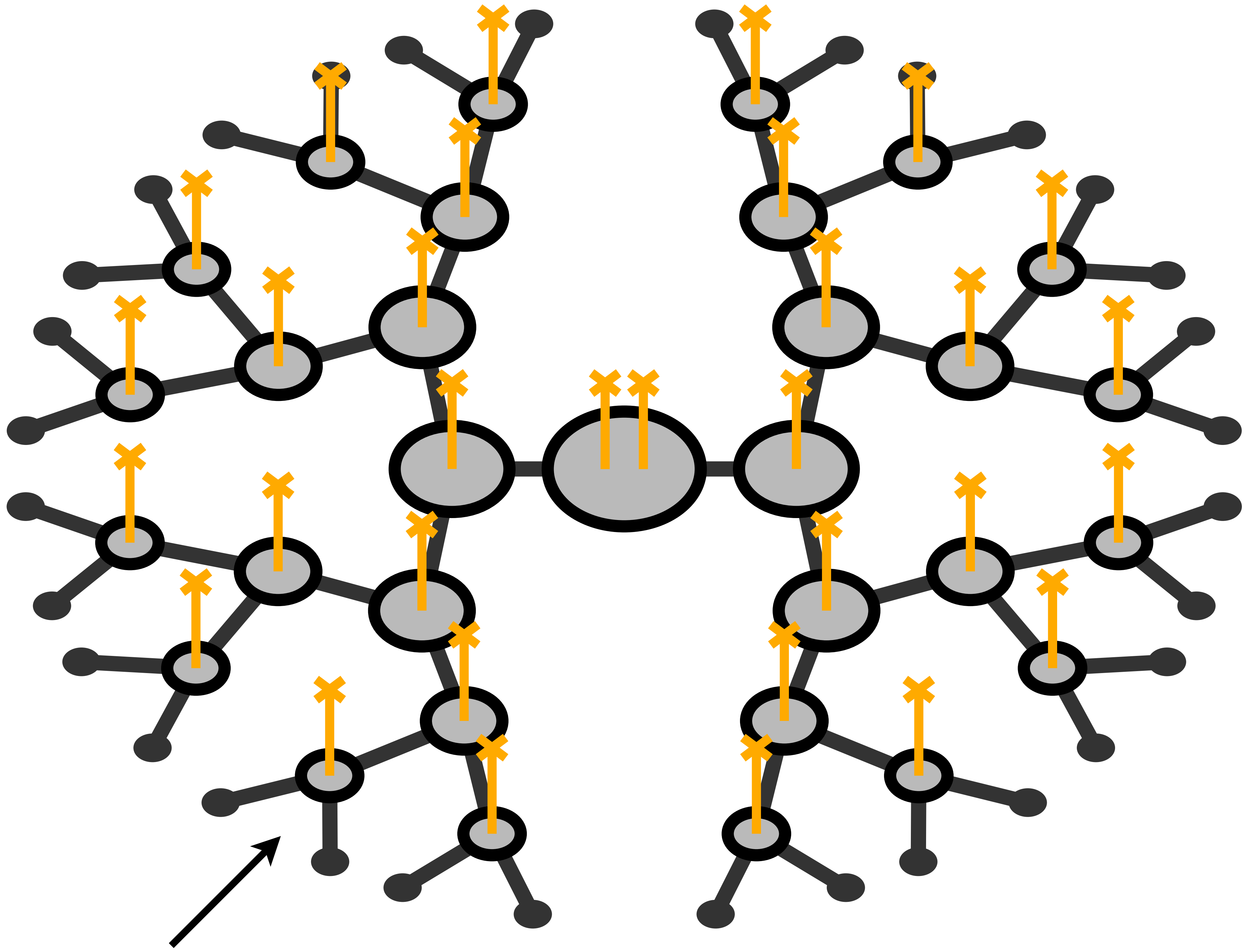




Nonlinear & adaptive generalizations of wavelets

Guy, Wavelets & RG1999+ White, Evenbly, Qi, Wavelets, MERA, and holographic mapping 2013+

Neural network holographic RG



Bijective neural net

Physical variables on the boundary

Latent variables in the bulk

RG flows along the radial direction

Information is preserved by the flow

Hu, Li, LW, You, PRR '20

See also Hashimoto et al 1809.10536, 2006.00712

Mutual information reveals the emergent geometry in the bulk

Continuous normalizing flows

$$\ln p(\boldsymbol{x}) = \ln \mathcal{N}(\boldsymbol{z}) - \ln \left| \det \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} \right) \right|$$

Consider infinitesimal change-of-variables Chen et al 1806.07366

$$\boldsymbol{x} = \boldsymbol{z} + \varepsilon \boldsymbol{v} \qquad \ln p(\boldsymbol{x}) - \ln \mathcal{N}(\boldsymbol{z}) = -\ln \left| \det \left(1 + \varepsilon \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{z}} \right) \right|$$

$\varepsilon \to 0$

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v} \qquad \frac{d \ln p(\boldsymbol{x}, t)}{dt} = -\nabla \cdot \boldsymbol{v}$$

Continuous normalizing flows

$$\ln p(\boldsymbol{x}) = \ln \mathcal{N}(\boldsymbol{z}) - \ln \left| \det \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} \right) \right|$$

Consider infinitesimal change-of-variables Chen et al 1806.07366

$$\boldsymbol{x} = \boldsymbol{z} + \varepsilon \boldsymbol{v} \qquad \ln p(\boldsymbol{x}) - \ln \mathcal{N}(\boldsymbol{z}) = -\ln \left| \det \left(1 + \varepsilon \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{z}} \right) \right|$$

$\varepsilon \to 0$ $\qquad t = 0 \qquad t = 1$

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v} \qquad \frac{d \ln p(\boldsymbol{x}, t)}{dt} = -\nabla \cdot \boldsymbol{v}$$

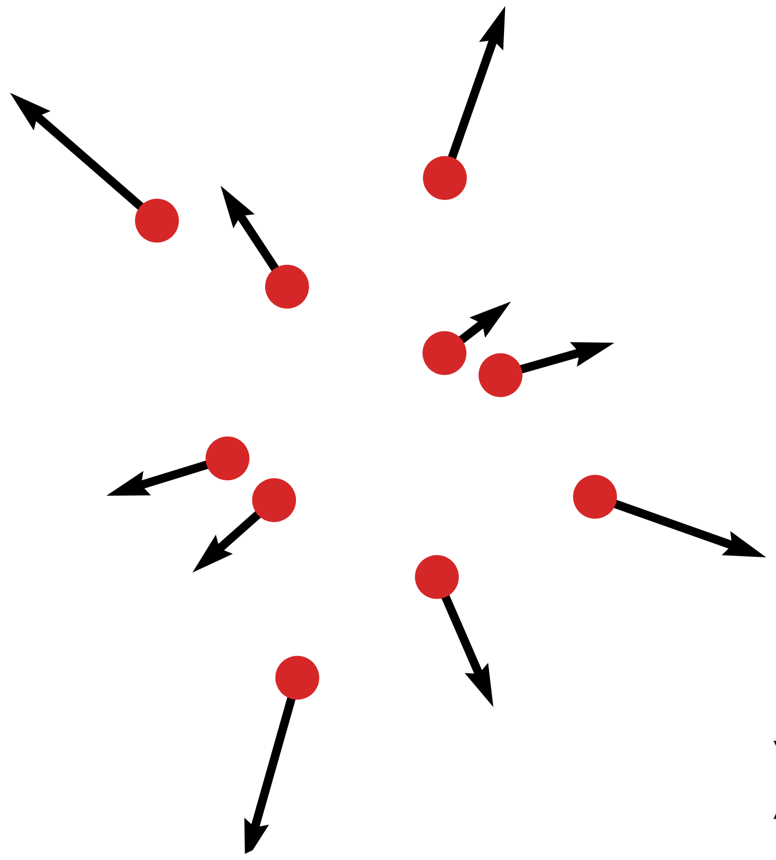
Fluid physics behind flows

Zhang, E, LW 1809.10188

wangleiphy/MongeAmpereFlow

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}$$

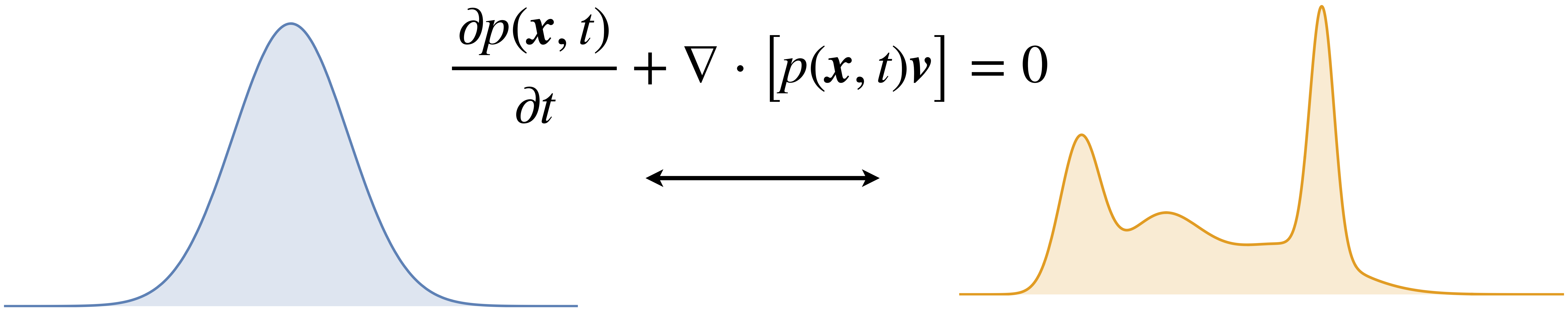
$$\frac{d \ln p(\boldsymbol{x}, t)}{dt} = -\nabla \cdot \boldsymbol{v}$$



$$\frac{d}{dt} = \frac{\partial}{\partial t} + \boldsymbol{v} \cdot \nabla$$ “material derivative”

Lagrangian v.s. Euler approach to fluid mechanics

$$\frac{\partial p(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x}, t)\boldsymbol{v}\right] = 0$$



Simple density

Complex density

Infinitesimal Flows Another way to reduce the computational overhead of normalizing flows is to use an ordinary differential equation to generate f [2]. In this case, the probability distribution changes over a finite time from $p(\mathbf{x}; 0)$ to $p(\mathbf{z}; T)$, where $\mathbf{z}$ is the end point of a curve defined by the ODE $\dot{\mathbf{x}}(t) = \mathbf{v}(\mathbf{x}(t))$, $\mathbf{x}(0) = \mathbf{x}$. For a small time step dt, we can approximate $\mathbf{x}(t + dt)$ to first order as $\mathbf{x}(t + dt) = \mathbf{x}(t) + dt\mathbf{v}(\mathbf{x}(t)) + \mathcal{O}(dt^2)$. Plugging this into Eq. 2 yields:

$$\begin{aligned}
\log p(\mathbf{x} + dt\mathbf{v}(\mathbf{x}) + \mathcal{O}(dt^2); t + dt) &= \log p(\mathbf{x}; t) - \log|\mathbf{J}_f(\mathbf{x})| && (3)\\
&= \log p(\mathbf{x}; t) - \log|\mathbf{I} + dt\mathbf{J}_{\mathbf{v}}(\mathbf{x}) + \mathcal{O}(dt^2)| && (4)
\end{aligned}$$

Taking a Taylor series gives:

$$\log p(\mathbf{x}; t + dt) + dt\mathbf{v}(\mathbf{x})^T \nabla \log p(\mathbf{x}; t + dt) = \log p(x; t) - dt\mathrm{Tr}(\mathbf{J}_v(\mathbf{x})) + \mathcal{O}(dt^2) \quad (5)$$

which, in the limit as $dt \to 0$, becomes:

$$\frac{\partial \log p(\mathbf{x}; t)}{\partial t} = -\mathbf{v}(\mathbf{x})^T \nabla \log p(\mathbf{x}; t) - \mathrm{Tr}(\mathbf{J}_{\mathbf{v}}(\mathbf{x})) = -\mathbf{v}^T \nabla \log p(\mathbf{x}; t) - \nabla \cdot \mathbf{v} \quad (6)$$

after some rearranging of terms. Here $\nabla\cdot$ is the divergence of a vector field, which is just another way of writing the trace of the Jacobian. The right-hand side of this equation is also the trace of the *Stein operator* of the distribution $p(\mathbf{x})$ applied to the function $\mathbf{v}(\mathbf{x})$, and plays a critical role in Stein variational gradient descent (SVGD) [13]. Switching from the log density to the density (and dropping the t for clarity), we find this expression can be simplified considerably:

$$\begin{aligned}
\frac{1}{p(\mathbf{x})}\frac{\partial p(\mathbf{x})}{\partial t} &= -\mathbf{v}^T \frac{\nabla p(\mathbf{x})}{p(\mathbf{x})} - \nabla \cdot \mathbf{v}\\
\frac{\partial p(\mathbf{x})}{\partial t} &= -\mathbf{v}^T \nabla p(\mathbf{x}) - p(\mathbf{x})\nabla \cdot \mathbf{v}\\
&= -\nabla \cdot (\mathbf{v}(\mathbf{x})p(\mathbf{x})) && (7)
\end{aligned}$$

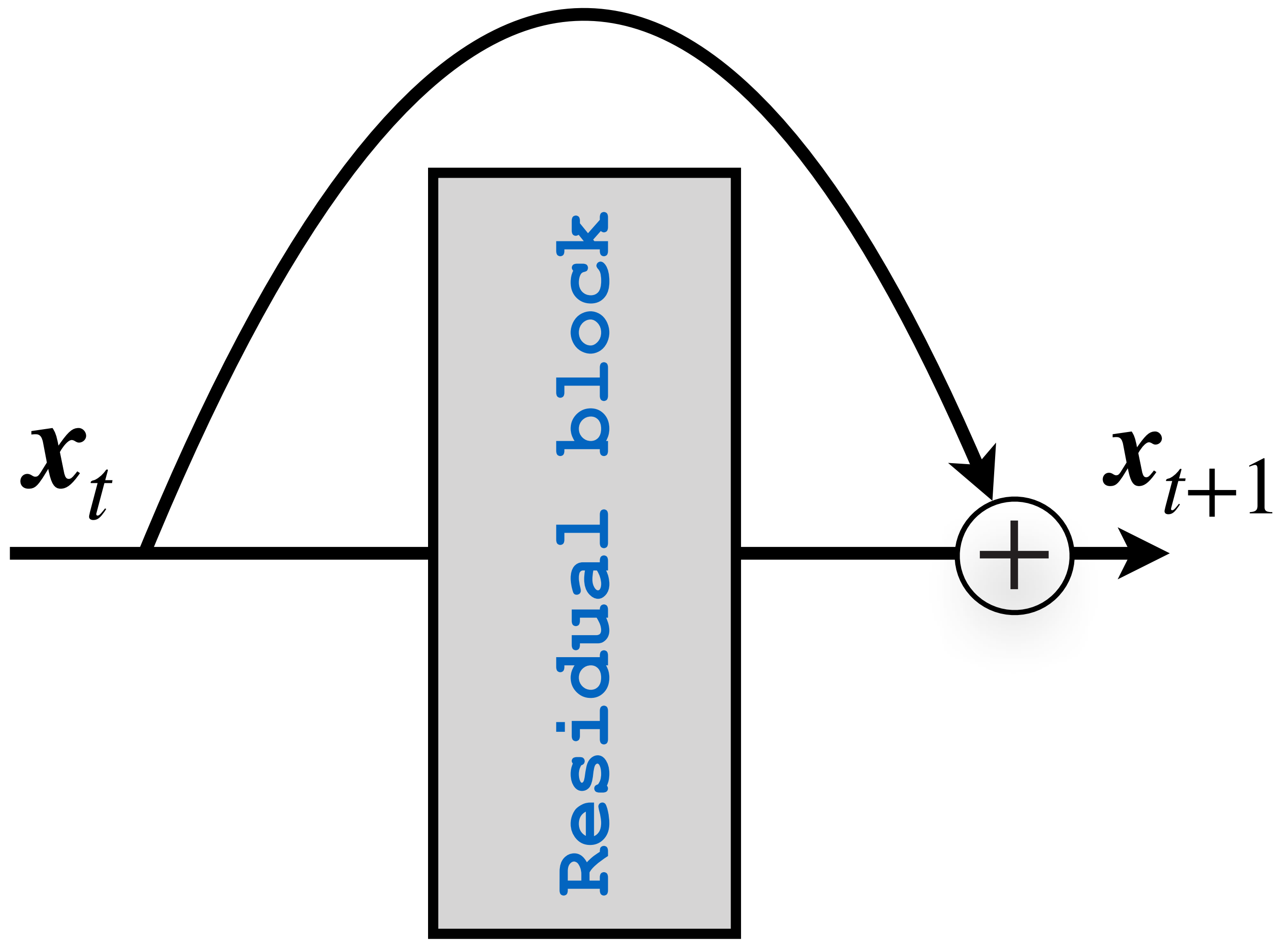
This may also be familiar as the drift term of the Fokker-Planck equation [11, Eq. 6.48] or the continuity equation for conservation of mass in fluid mechanics. We will denote the change to a

Pfau & Rezende

2012.02035

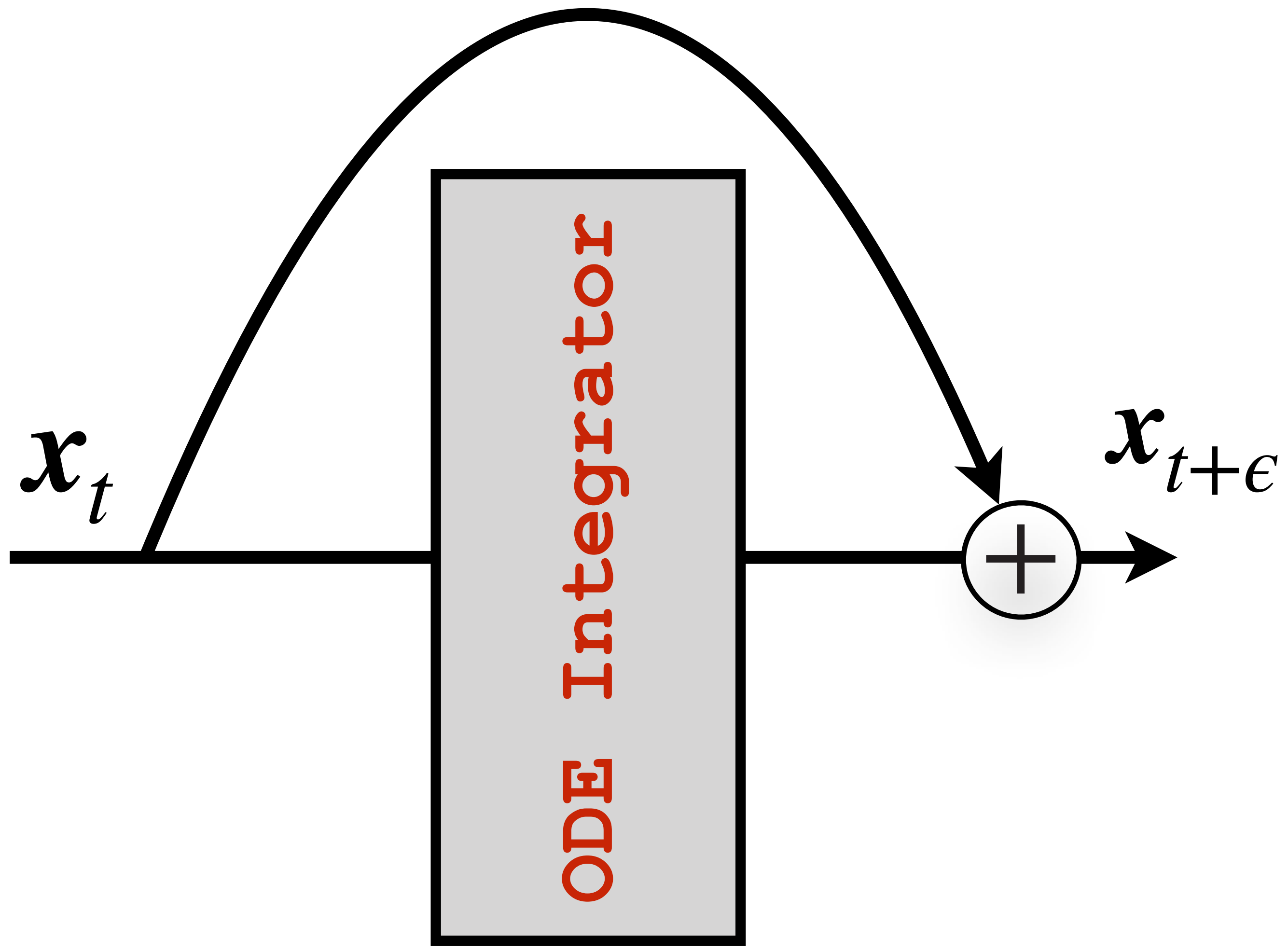
Neural Ordinary Differential Equations

Residual network



$$x_{t+1} = x_t + v(x_t)$$

ODE integration

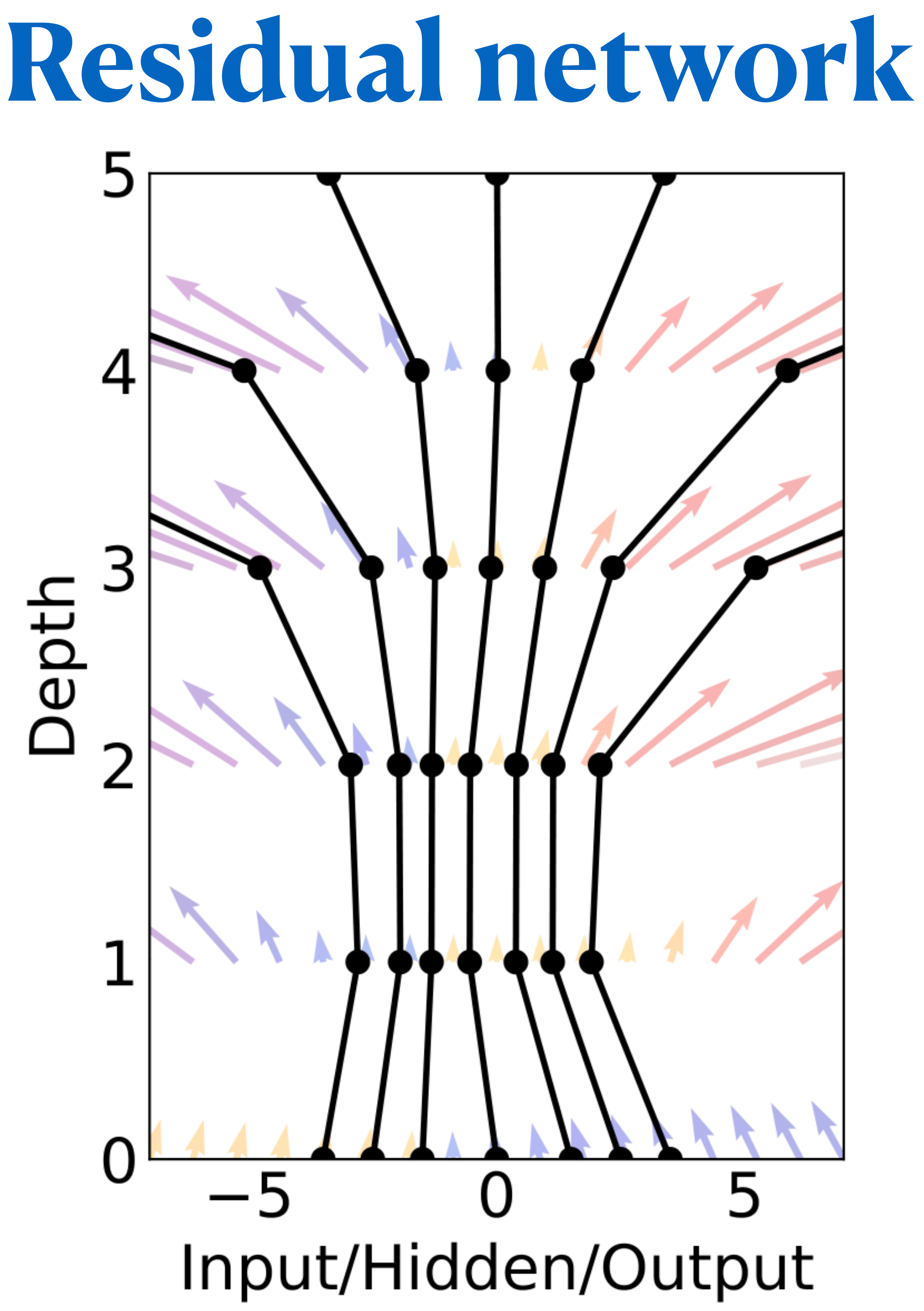


$$dx/dt = v(x)$$

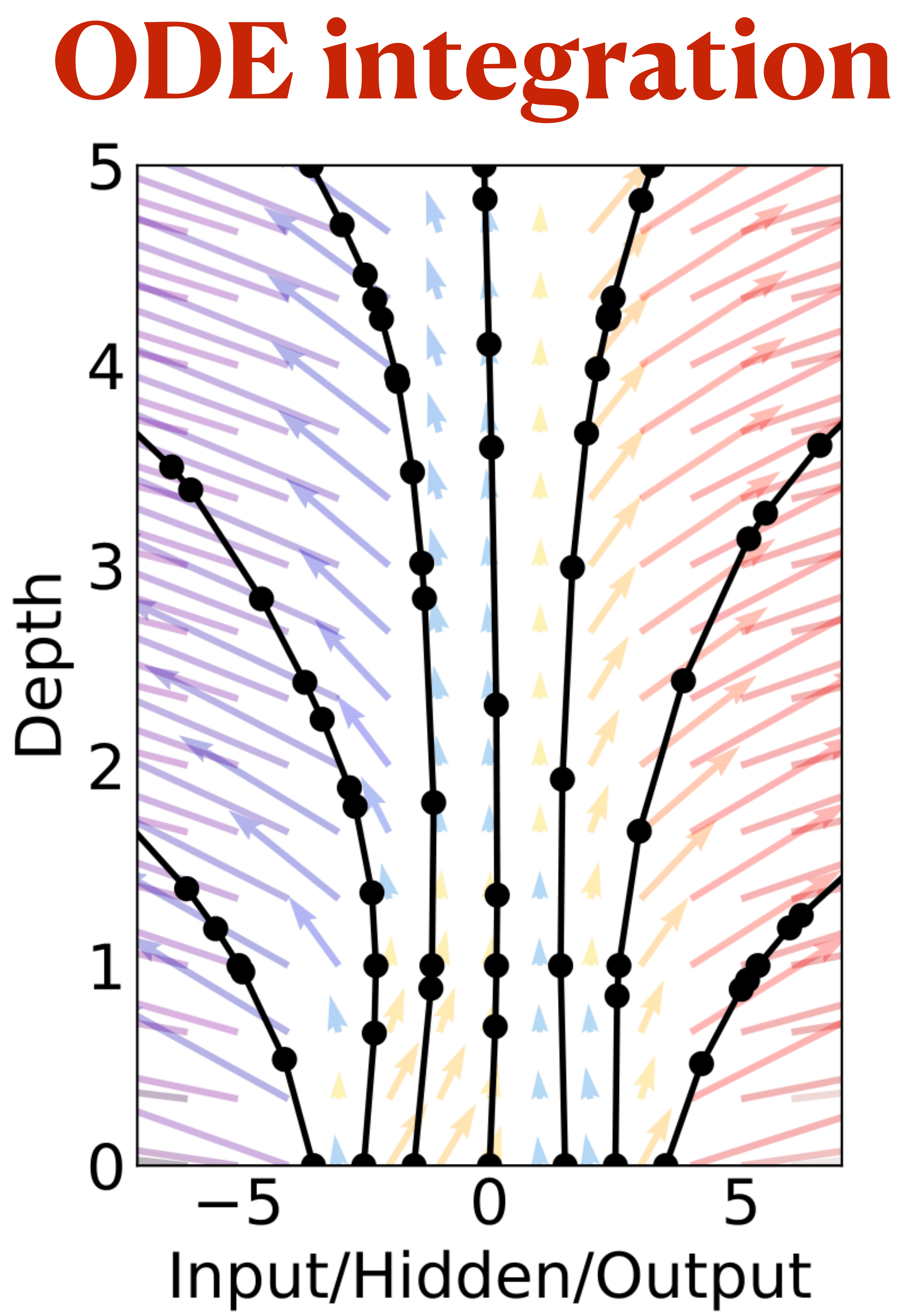
Chen et al, 1806.07366

Harbor el al 1705.03341
Lu et al 1710.10121,
E Commun. Math. Stat 17'...

Neural Ordinary Differential Equations



$$x_{t+1} = x_t + v(x_t)$$



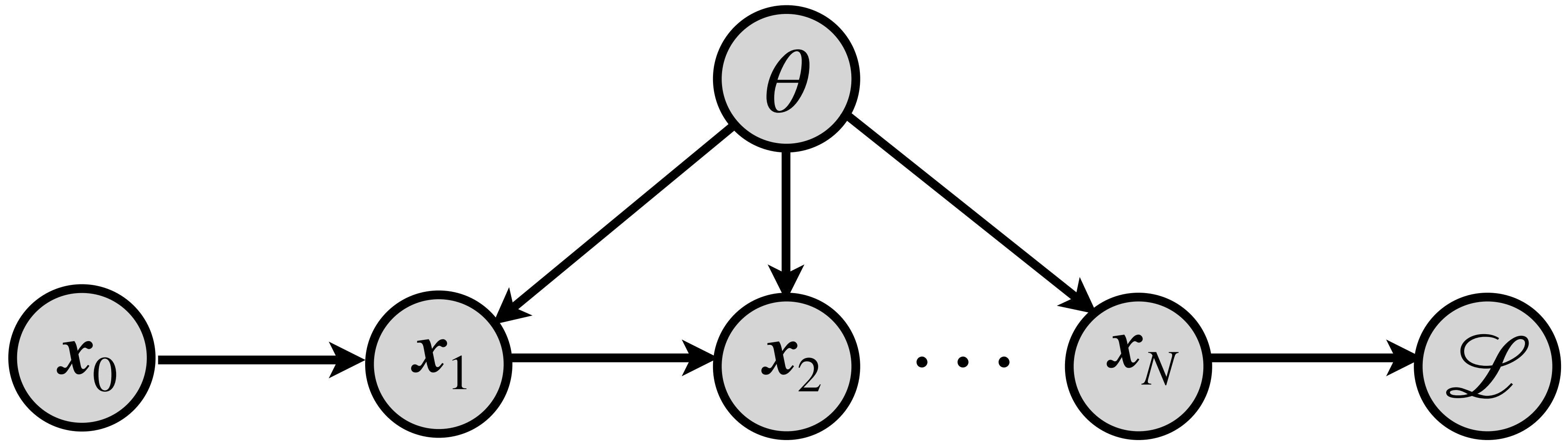
$$dx/dt = v(x)$$

Harbor el al 1705.03341
Lu et al 1710.10121,
E Commun. Math. Stat 17'...

Chen et al, 1806.07366

Backpropagate through an ODE

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\theta}, t)$$



Adjoint $\bar{\boldsymbol{x}}(t) = \dfrac{\partial \mathscr{L}}{\partial \boldsymbol{x}(t)}$ satisfies another ODE to be integrated back in time

$$\frac{d\bar{\boldsymbol{x}}(t)}{dt} = -\,\bar{\boldsymbol{x}}(t)\frac{\partial \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\theta}, t)}{\partial \boldsymbol{x}}$$

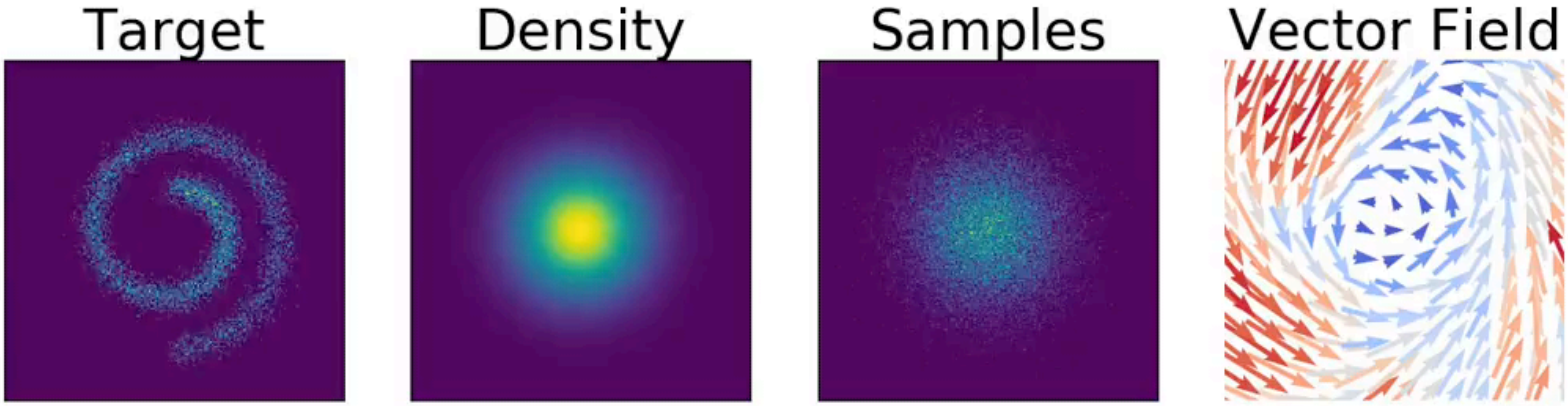
Gradient w.r.t. parameter

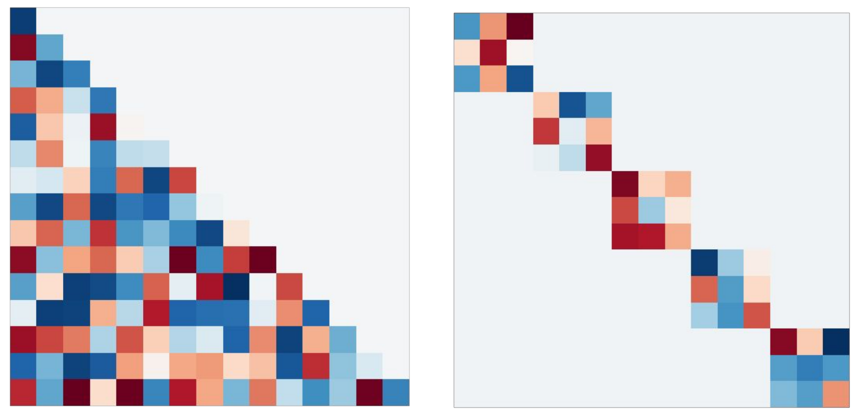
$$\frac{\partial \mathscr{L}}{\partial \boldsymbol{\theta}} = \int_0^T dt\, \bar{\boldsymbol{x}}(t)\frac{\partial \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\theta}, t)}{\partial \boldsymbol{\theta}}$$

Exercise:
Derive this!

Continuous normalizing flows implemented with NeuralODE

Chen et al, 1806.07366, Grathwohl et al 1810.01367

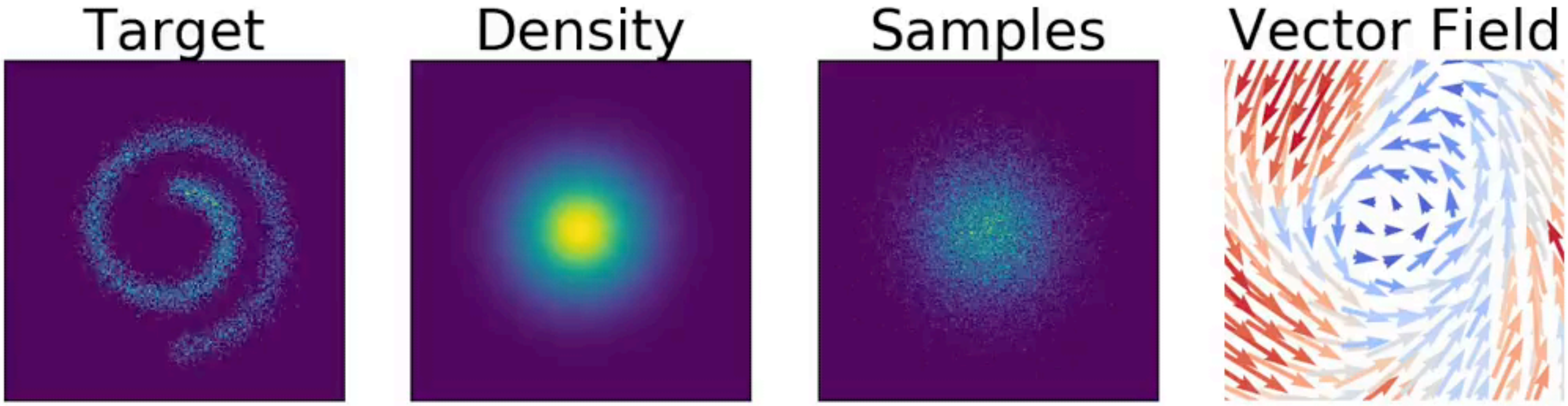


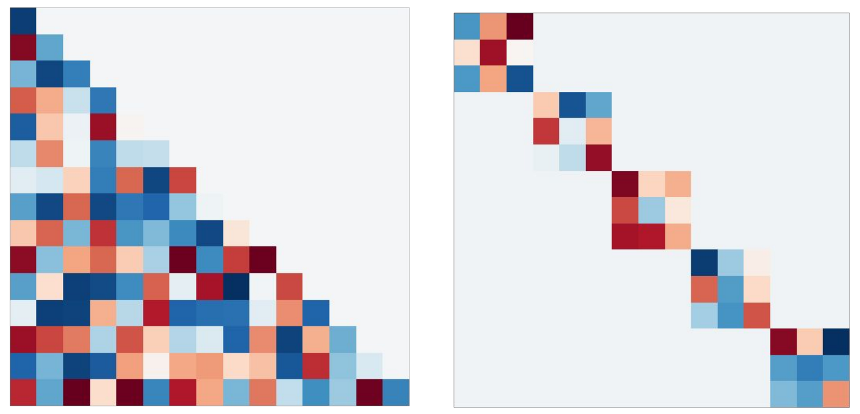


Continuous normalizing flow have no structural constraints on the transformation Jacobian

Continuous normalizing flows implemented with NeuralODE

Chen et al, 1806.07366, Grathwohl et al 1810.01367





Continuous normalizing flow have no structural constraints on the transformation Jacobian

The two use cases

Zhang, E, LW, 1809.10188

Maximum likelihood estimation

$$\mathcal{N}(\boldsymbol{x}) \xleftarrow{\int_T^0 \dot{\boldsymbol{x}}\, dt}$$

"learn from data"

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

Variational free energy

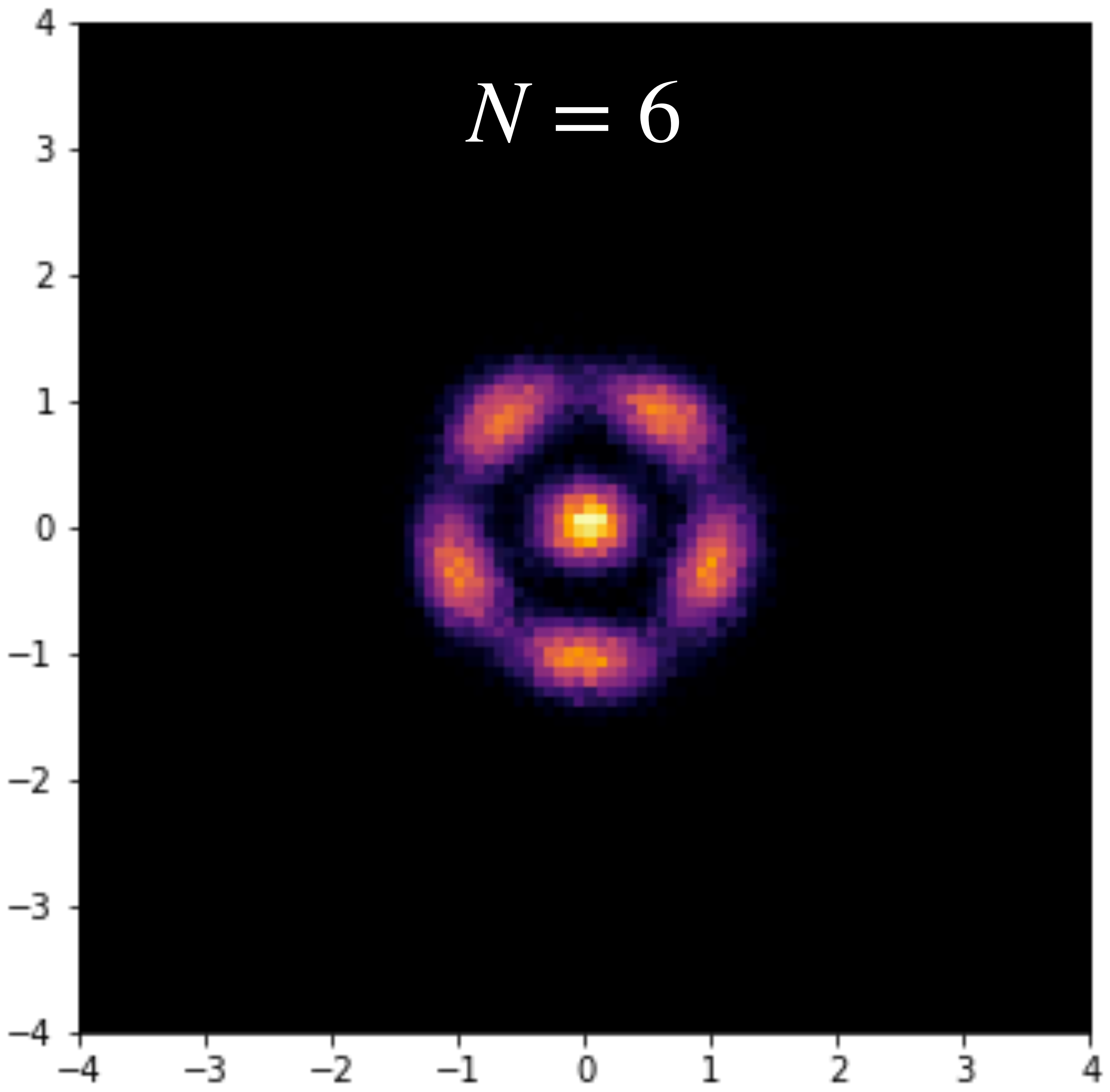
$$\mathcal{N}(\boldsymbol{x}) \xrightarrow{\int_0^T \dot{\boldsymbol{x}}\, dt} \frac{e^{-E(\boldsymbol{x})}}{Z}$$

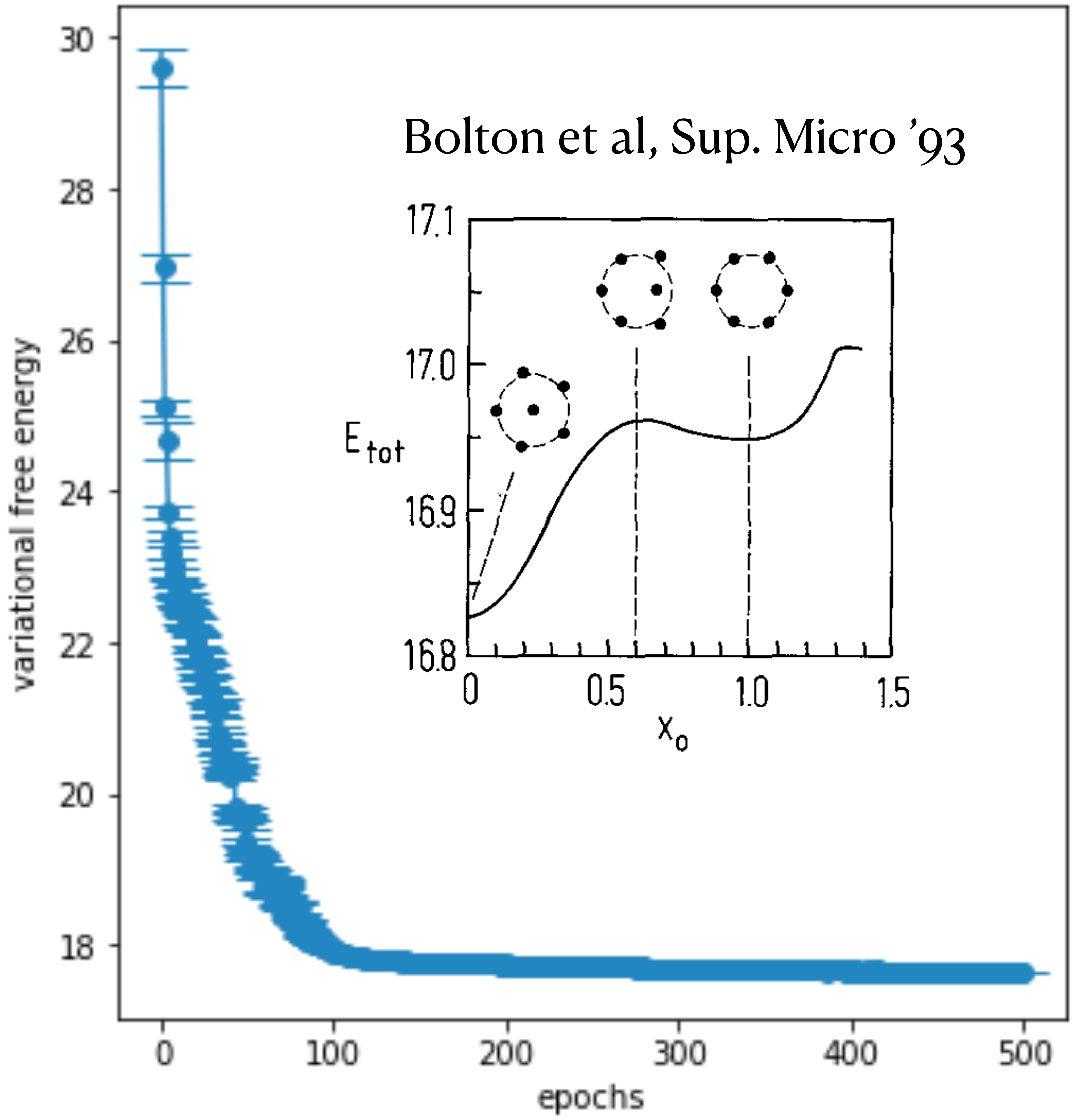
"learn from Energy"

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

Demo: Classical Coulomb gas in a harmonic trap

$$E = \sum_{i<j} \frac{1}{|\boldsymbol{x}_i - \boldsymbol{x}_j|} + \sum_i^N \boldsymbol{x}_i^2$$

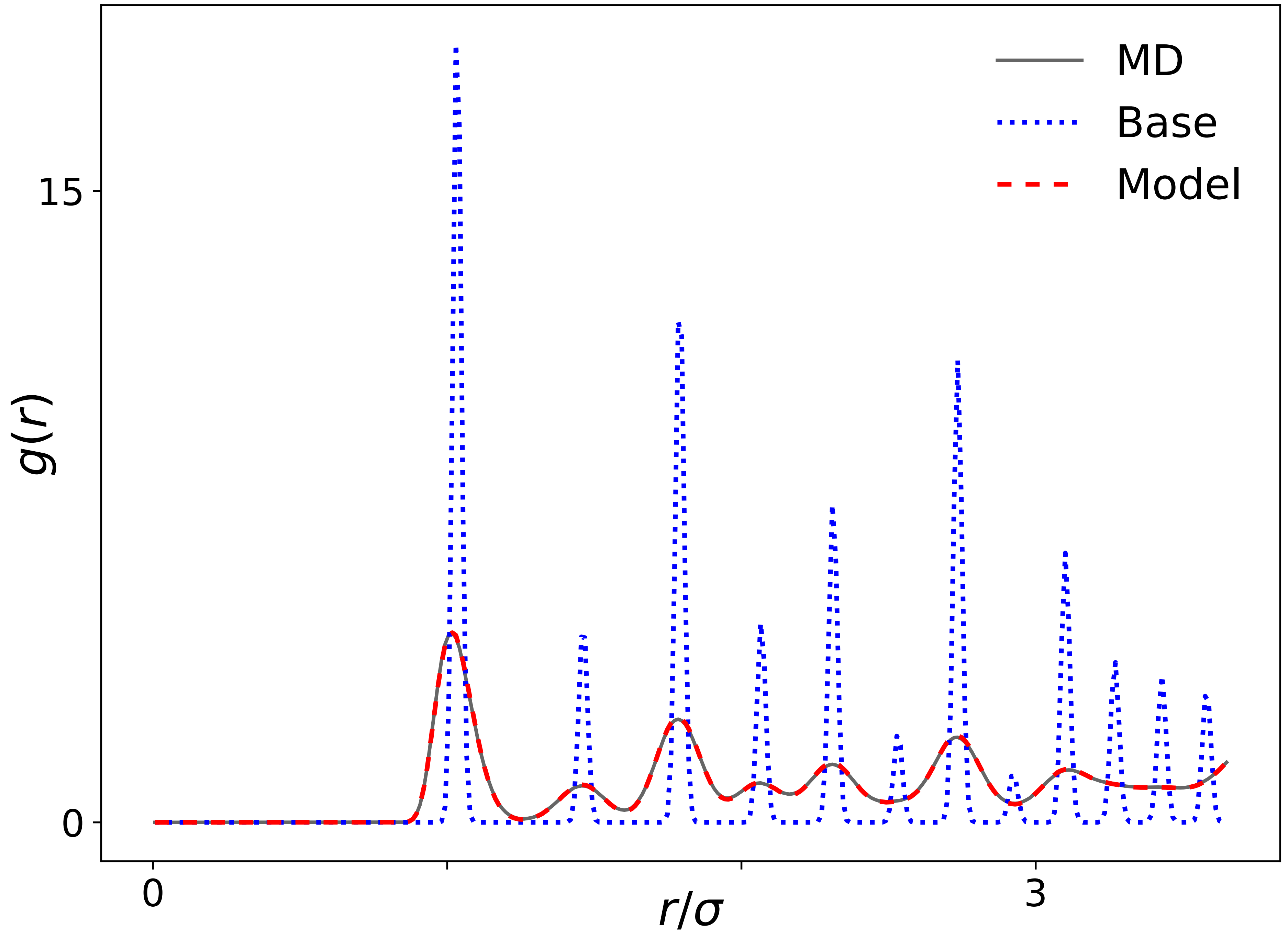




https://colab.research.google.com/drive/1yIlPo5CAjYrqWHeFEZrMlzWNCoNJ6_YP#scrollTo=eQwLElKmaowu

Case study: Normalizing flow for atomic solids

Variational free energy with a really deep permutation equivariant flow



System	N	LFEP	LBAR	MBAR
LJ	256	3.10800(28)	3.10797(1)	3.10798(9)
LJ	500	3.12300(41)	3.12264(2)	3.12262(10)

$$\ln Z = \ln \mathbb{E}_{x\sim q(x)}\left[e^{-\beta E(x) - \ln q(x)}\right]$$

free energy perturbation (Zwanzig 1954)

$$\ln Z_B - \ln Z_A = \ln \mathbb{E}_A\left[e^{-\beta(E_B - E_A)}\right]$$

Wirnsberger et al, 2111.08696 https://github.com/deepmind/flows_for_atomic_solids

Normalizing flow for atomic solids

F. Hardware details and computational cost

For our flow experiments, we used 16 A100 GPUs to train each model on the bigger systems (512-particle mW and 500-particle LJ). It took approximately 3 weeks of training to reach convergence of the free-energy estimates. Obtaining 2M samples for evaluation took approximately 12 hours on 8 V100 GPUs for each of these models.

For each baseline MBAR estimate, we performed 100 separate simulations for LJ and 200 for mW, corresponding to the number of stages employed. These simulations were performed with LAMMPS [8] and each of them ran on multiple CPU cores communicating via MPI. We used 4 cores for the 64-particle and 216-particle mW experiments and 8 cores for all other systems. The MD simulations completed after approximately 11 and 14 hours for LJ (256 and 500 particles), and 7, 20 and 48 hours for mW (64, 216 and 512 particles). To evaluate the energy matrix for a single MBAR

Heavy lifting is mostly due to back-and-forth simulation of deep equivariant flow

Training: Monte Carlo Gradient Estimators

Review: 1906.10652

Reinforcement learning
Variational inference
Variational Monte Carlo
Variational quantum algorithms
...

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{\theta}}} \left[f(\boldsymbol{x}) \right]$$

Score function estimator (REINFORCE)

$$\nabla_{\theta} \mathbb{E}_{\boldsymbol{x} \sim p_{\theta}} \left[f(\boldsymbol{x}) \right] = \mathbb{E}_{\boldsymbol{x} \sim p_{\theta}} \left[f(\boldsymbol{x}) \, \nabla_{\theta} \ln p_{\theta}(\boldsymbol{x}) \right]$$

Pathwise estimator (Reparametrization trick) $\boldsymbol{x} = g_{\theta}(\boldsymbol{z})$

$$\nabla_{\theta} \mathbb{E}_{\boldsymbol{x} \sim p_{\theta}} \left[f(\boldsymbol{x}) \right] = \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z})} \left[\nabla_{\theta} f(g_{\theta}(\boldsymbol{z})) \right]$$

10.1 Guidance in Choosing Gradient Estimators

With so many competing approaches, we offer our rules of thumb in choosing an estimator, which follow the intuition we developed throughout the paper:

- If our estimation problem involves continuous functions and measures that are continuous in the domain, then using the pathwise estimator is a good default. It is relatively easy to implement and a default implementation, one without other variance reduction, will typically have variance that is low enough so as not to interfere with the optimisation.
- If the cost function is not differentiable or a black-box function then the score-function or the measure-valued gradients are available. If the number of parameters is low, then the measure-valued gradient will typically have lower variance and would be preferred. But if we have a high-dimensional parameter set, then the score function estimator should be used.
- If we have no control over the number of times we can evaluate a black-box cost function, effectively only allowing a single evaluation of it, then the score function is the only estimator of the three we reviewed that is applicable.
- The score function estimator should, by default, always be implemented with at least a basic variance reduction. The simplest option is to use a baseline control variate estimated with a running average of the cost value.
- When using the score-function estimator, some attention should be paid to the dynamic range of the cost function and its variance, and to find ways to keep its value bounded within a reasonable range, e.g., transforming the cost so that it is zero mean, or using a baseline.
- For all estimators, track the variance of the gradients if possible and address high variance by using a larger number of samples from the measure, decreasing the learning rate, or clipping the gradient values. It may also be useful to restrict the range of some parameters to avoid extreme values, e.g., by clipping them to a desired interval.
- The measure-valued gradient should be used with some coupling method for variance reduction. Coupling strategies that exploit relationships between the positive and negative components of the density decomposition, and which have shared sampling paths, are known for the commonly-used distributions.
- If we have several unbiased gradient estimators, a convex combination of them might have lower variance than any of the individual estimators.
- If the measure is discrete on its domain then the score-function or measure-valued gradient are available. The choice will again depend on the dimensionality of the parameter space.
- In all cases, we strongly recommend having a broad set of tests to verify the unbiasedness of the gradient estimator when implemented.

Mohamed et al, 1906.10652

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}}\left[f(\boldsymbol{x})\right]$$

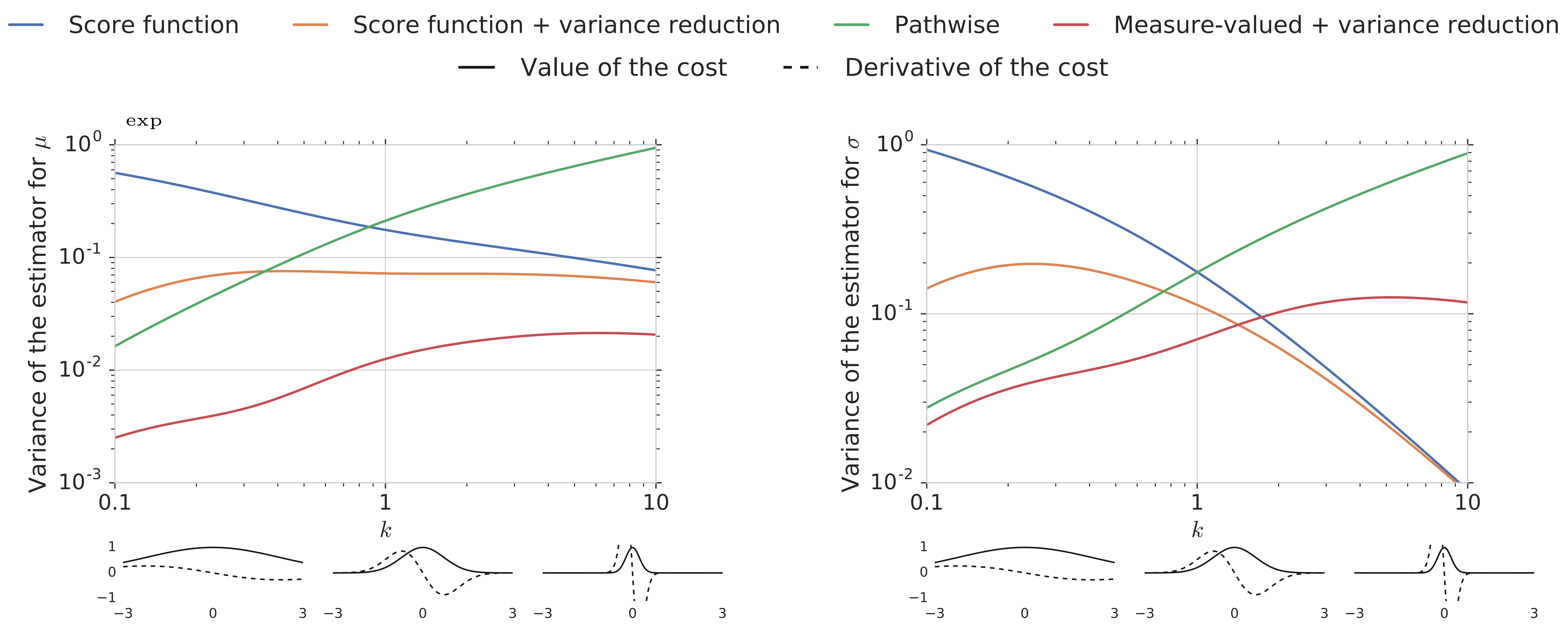
When to use which ?

More discussions

Roeder et al, 1703.09194

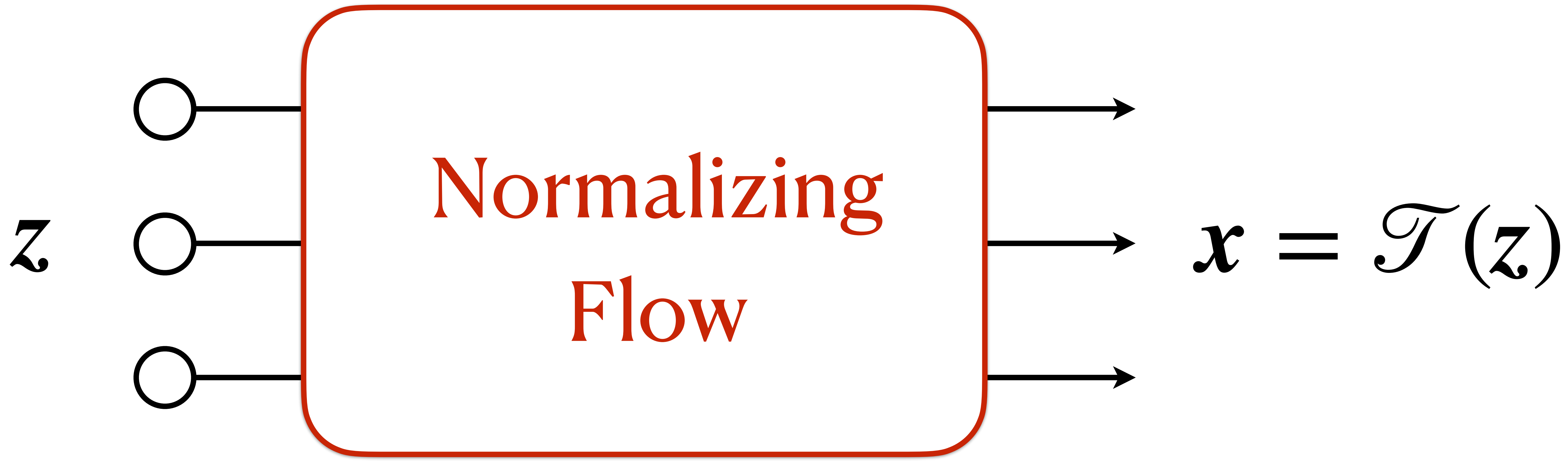
Vaitl et al 2206.09016, 2207.08219

$$\eta = \nabla_\theta \int \mathcal{N}(x|\mu, \sigma^2) f(x; k) dx; \quad \theta \in \{\mu, \sigma\}$$



https://github.com/deepmind/mc_gradients Mohamed et al, 1906.10652

Symmetries



Invariance

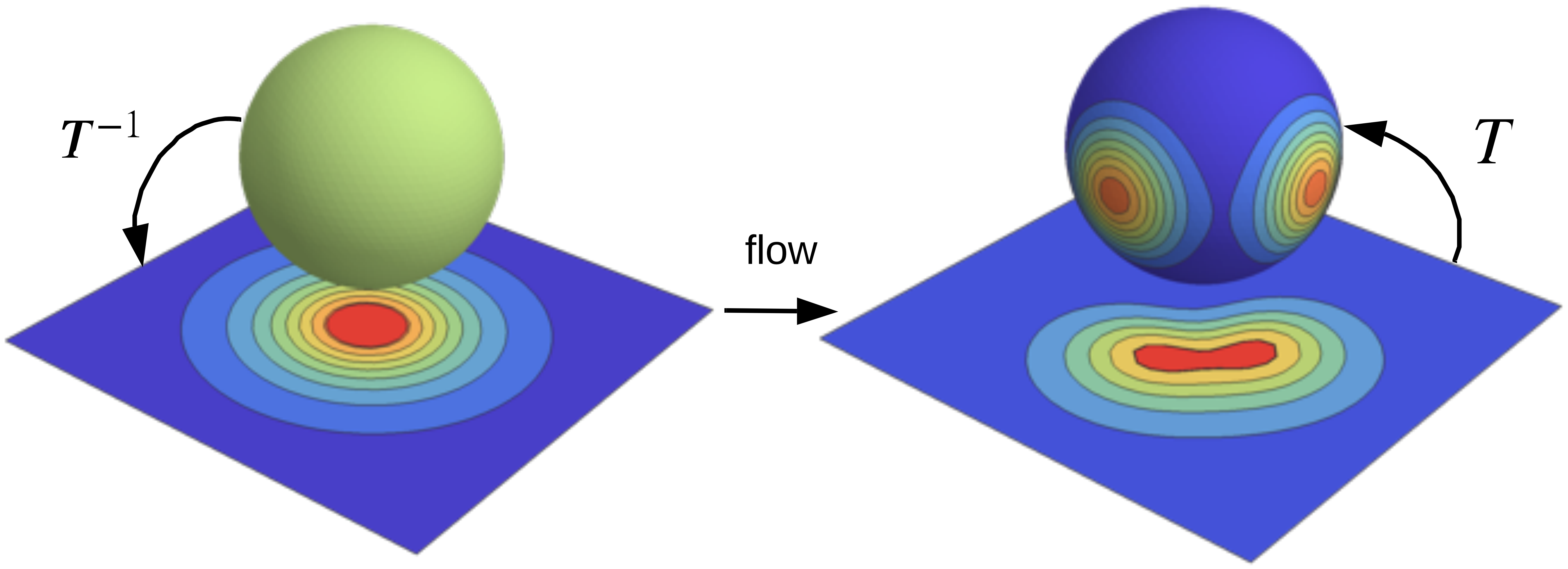
$$\rho(g\,\boldsymbol{x}) = \rho(\boldsymbol{x})$$

Equivariance

$$\mathcal{T}(g\,z) = g\mathcal{T}(z)$$

Spatial symmetries, permutation symmetries, gauge symmetries...

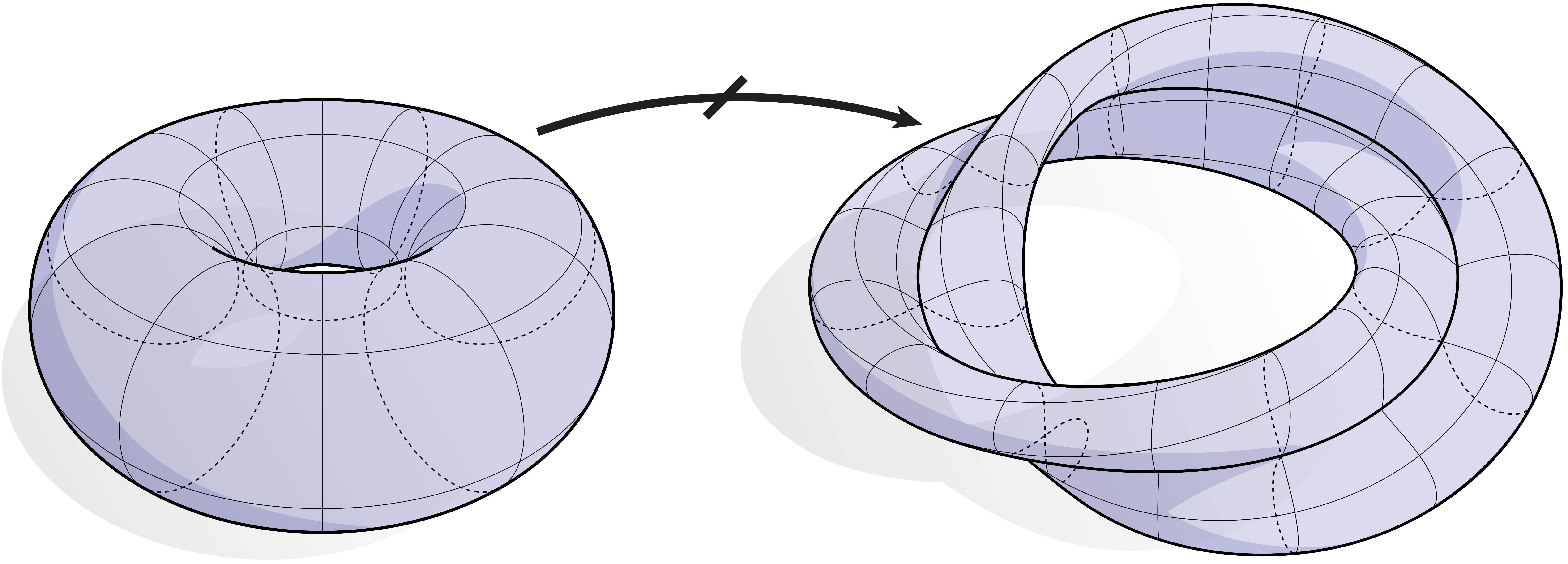
Flow on manifolds



Periodic variables, gauge fields, ...

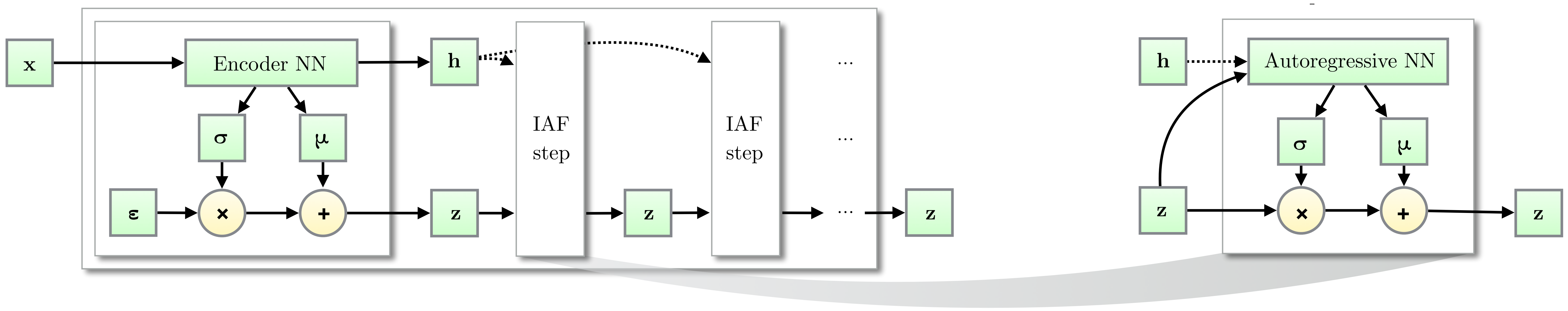
Gemici et al 1611.02304, Rezende et al, 2002.02428, Boyda et al, 2008.05456
Neural ODE on manifolds, Falorsi et al, 2006.06663, Lou et al, 2006.10254, Mathieu et al, 2006.10605

Obstructions

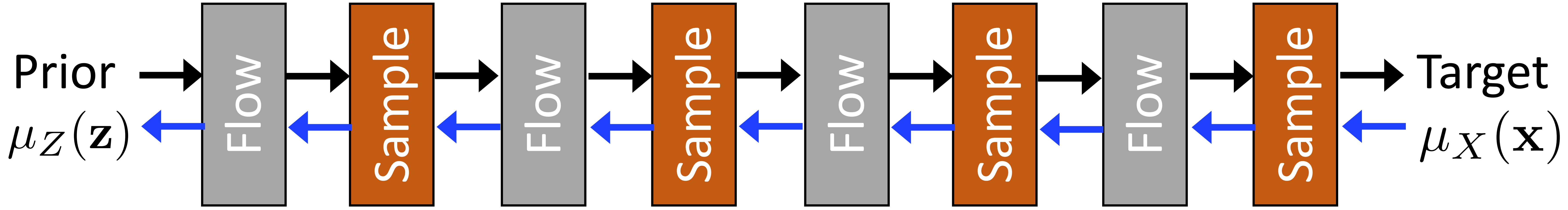


Dupont et al 1904.01681, Cornish et al, 1909.13833, Zhang et al, 1907.12998, Zhong et al, 2006.00392...

Mix with other approaches



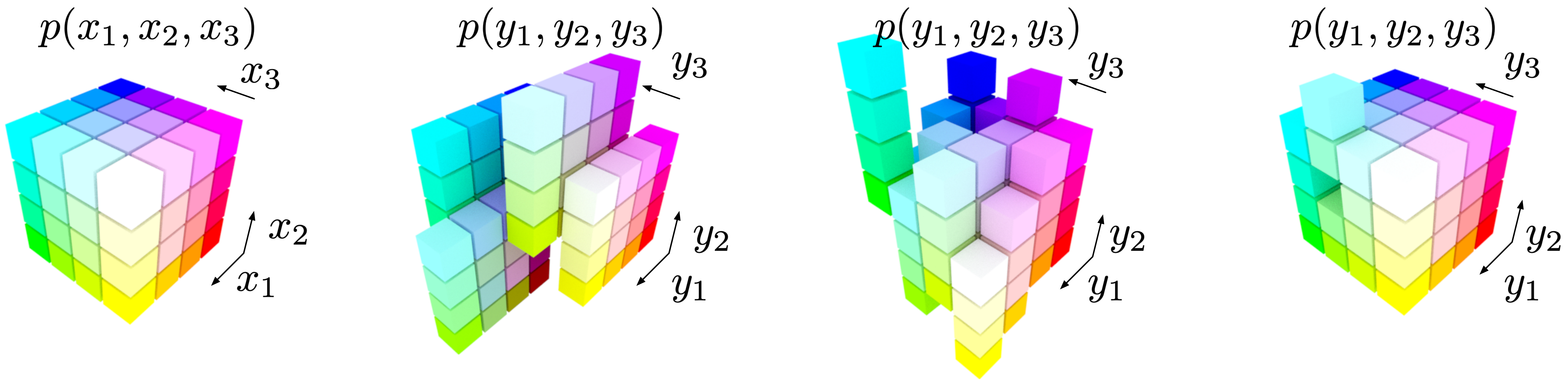
Kingma et al, 1606.04934,...



Levy et al, 1711.09268, Wu et al 2002.06707, ...

Discrete flows

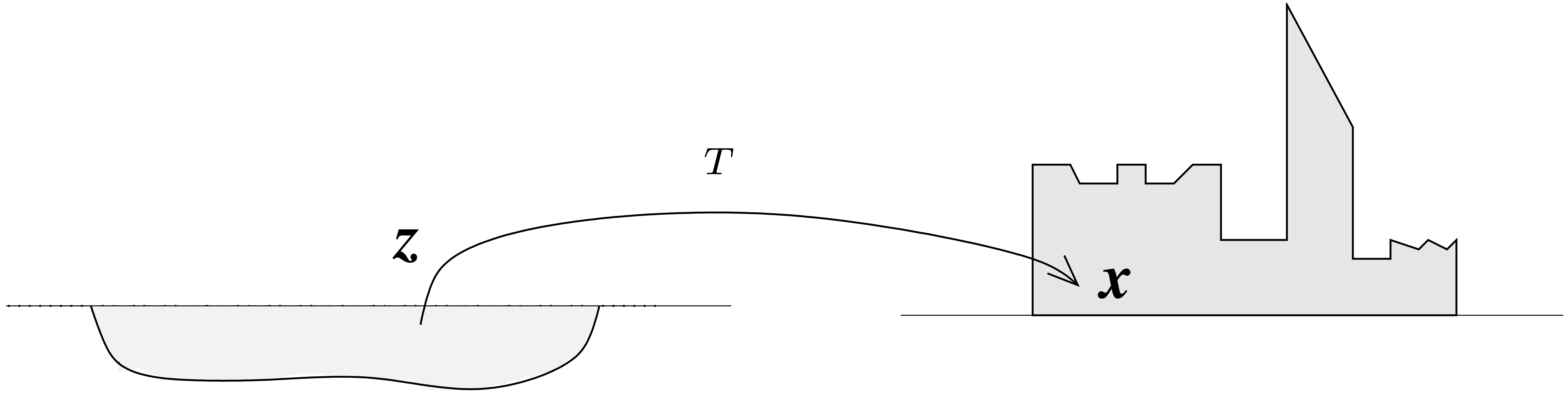
$$p(\boldsymbol{x}) = p(\boldsymbol{y} = \mathcal{T}(\boldsymbol{x}))$$



Tran et al, 1905.10347, Hoogeboom et al, 1905.07376, van den Berg 2006.12459

Optimal Transport Theory

Monge problem (1781): How to transport earth with optimal cost ?




Monge


Kantorovich


Koopmans


Dantzig


Brenier


Otto


McCann


Villani


Figalli

Nobel Prize in Economics '75

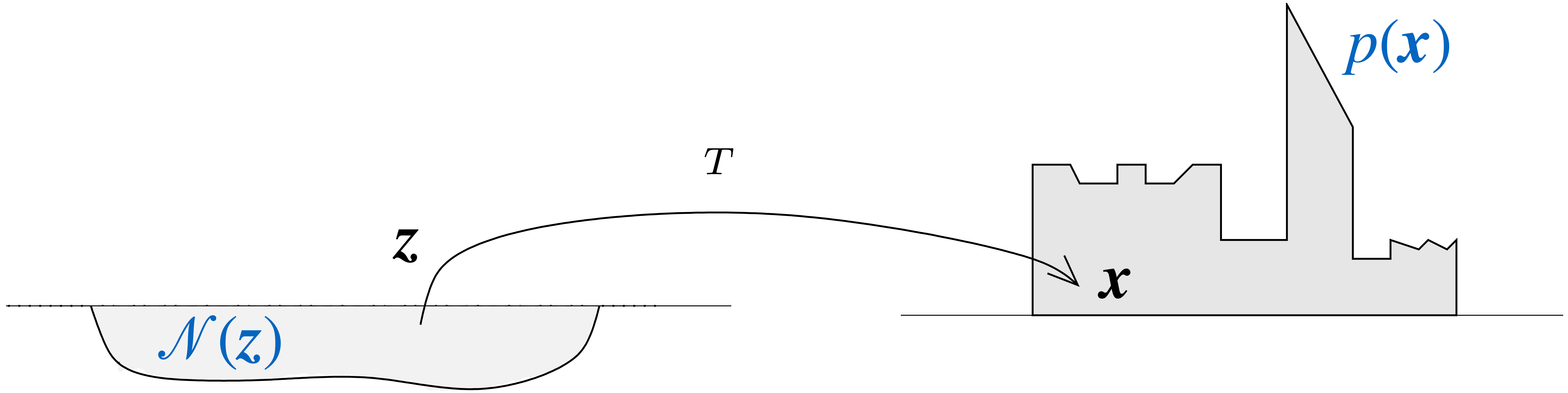
Fields Metal '10

Fields Metal '18

from Cuturi, Solomon NISP 2017 tutorial

Optimal Transport Theory

Monge problem (1781): How to transport earth with optimal cost ?




Monge


Kantorovich


Koopmans


Dantzig


Brenier


Otto


McCann


Villani


Figalli

Nobel Prize in Economics ’75

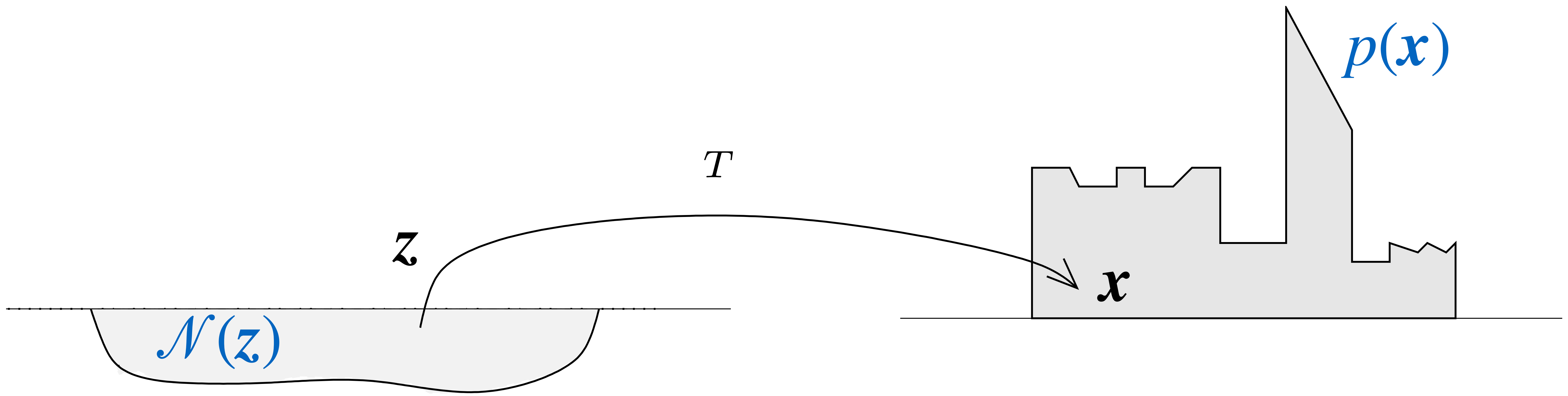
Fields Metal ’10

Fields Metal ’18

from Cuturi, Solomon NISP 2017 tutorial

Optimal Transport Theory

Monge problem (1781): How to transport earth with optimal cost ?



Brenier theorem (1991)

Under certain conditions
the optimal map is $\boxed{\boldsymbol{z} \mapsto \boldsymbol{x} = \nabla u(\boldsymbol{z})}$

Monge-Ampère Equation

$$\boxed{\frac{\mathcal{N}(z)}{p(\nabla u(z))} = \det\left(\frac{\partial^2 u}{\partial z_i \partial z_j}\right)}$$

Monge-Ampère Flow

Zhang, E, LW 1809.10188

wangleiphy/MongeAmpereFlow

$$\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot \left[\rho(\boldsymbol{x}, t) \nabla \varphi\right] = 0$$

1. Drive the flow with an “irrotational” velocity field

2. Impose symmetry to the scalar valued potential for symmetric generative model

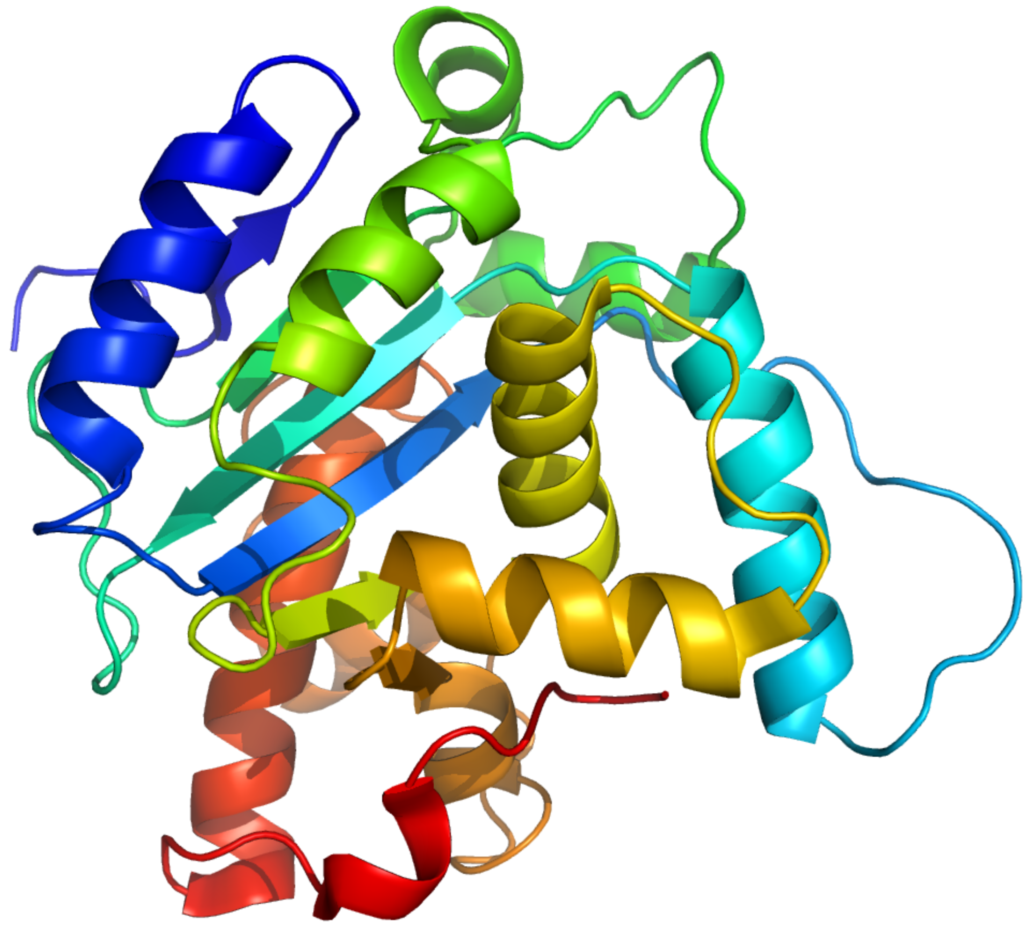
$$\varphi(g\,\boldsymbol{x}) = \varphi(\boldsymbol{x}) \quad \Longrightarrow \quad \rho(g\,\boldsymbol{x}) = \rho(\boldsymbol{x})$$

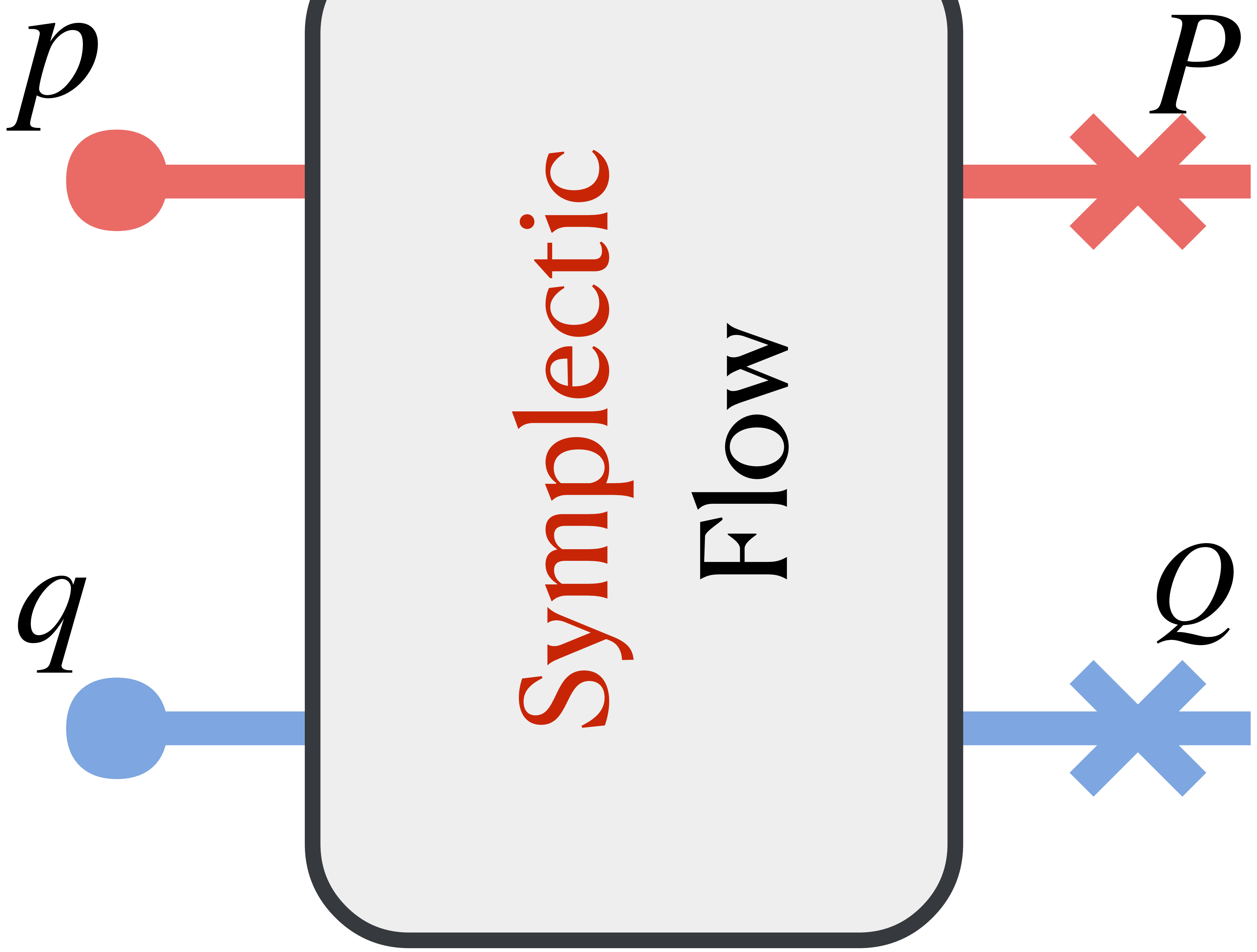
Neural Canonical Transformations

Li, Dong, Zhang, LW, PRX '20 [li012589/neuralCT]

$H(p,q)$

$$K(P,Q) = \sum_k \frac{P_k^2 + \omega_k^2 Q_k^2}{2}$$





physical space

latent space

Learn harmonic frequencies of the base to identify slow collective modes

See Bondesan et al 1906.04645, Ishikawa et al 2103.00372 for investigations on integrability

Generative models and their physics genes

Goodfellow, NIPS tutorial, 1701.00160

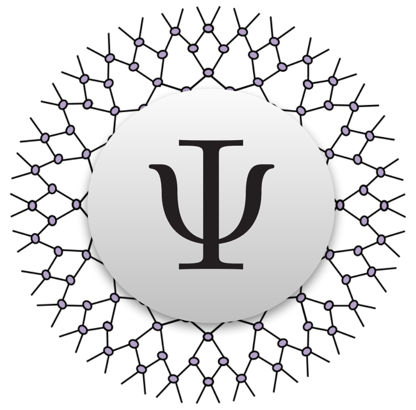
$p(\boldsymbol{x})$

Explicit density

Implicit density

Direct

GAN



Tensor Networks

Tractable density

-Fully visible belief nets

-NADE

-MADE

-PixelRNN

-Change of variables models (nonlinear ICA)

Approximate density

Variational

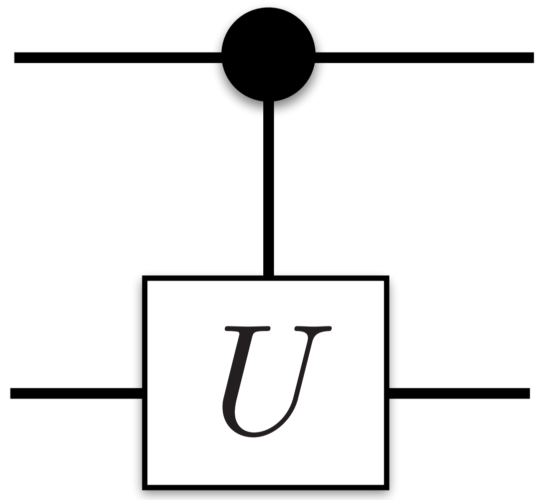
Variational autoencoder

Markov Chain

Boltzmann machine

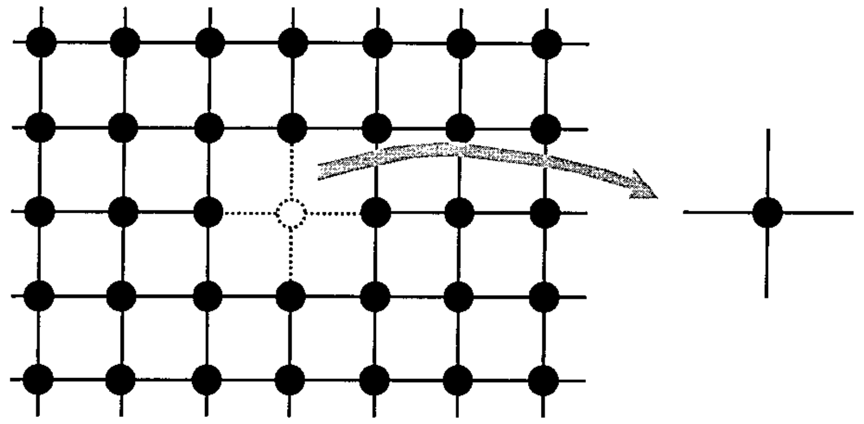
Markov Chain

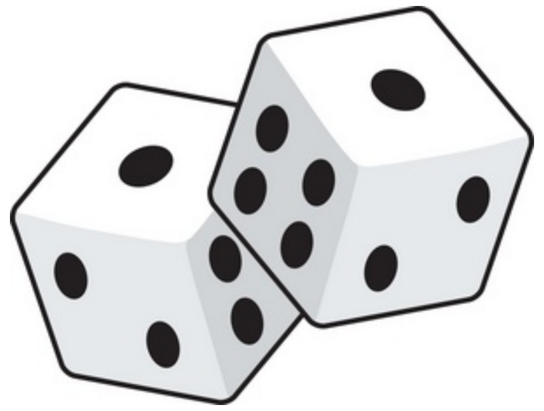
GSN

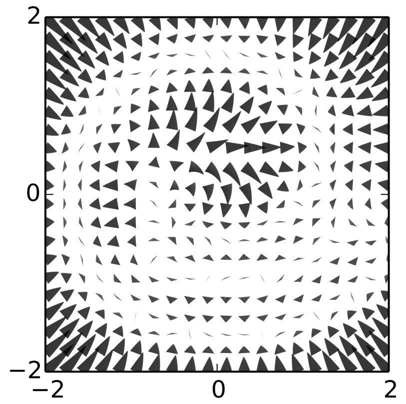


Quantum Circuits

+**Diffusion models**



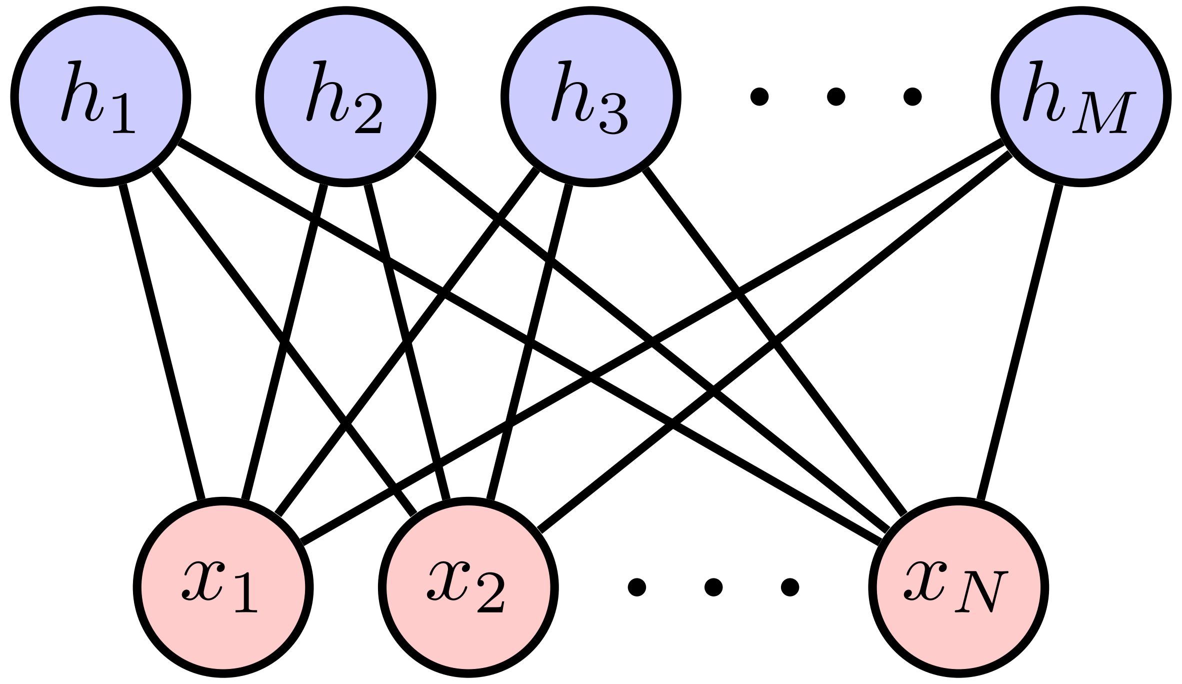




Boltzmann machines

$$\mathscr{L} = -\,\mathbb{E}_{\boldsymbol{x}\sim\text{data}}\left[\ln p(\boldsymbol{x})\right] \qquad p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})}/Z$$

Learn



$$\nabla_\theta \mathscr{L} = \langle \nabla_\theta E \rangle_{\text{data}} - \langle \nabla_\theta E \rangle_{\text{model}}$$

2210.10318

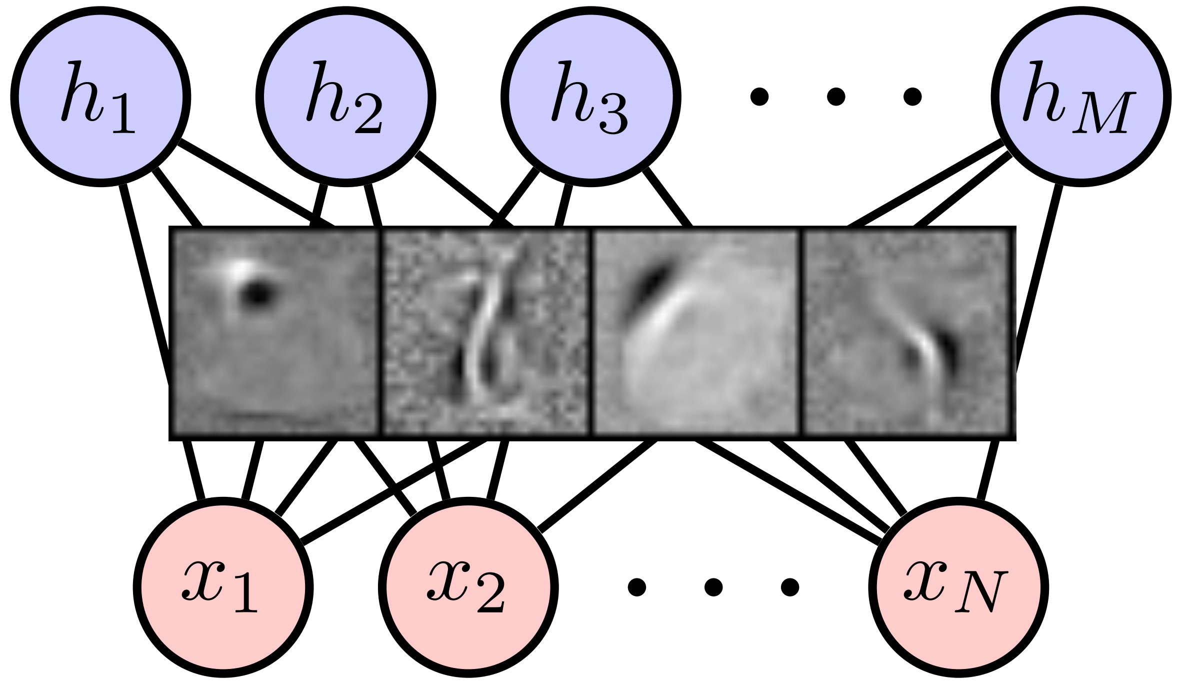
GAUSSIAN-BERNOULLI RBMS WITHOUT TEARS

Renjie Liao*[1], Simon Kornblith[2], Mengye Ren[3], David J. Fleet[2,4,5], Geoffrey Hinton[2,4,5]

Boltzmann machines

$$\mathscr{L} = -\,\mathbb{E}_{\boldsymbol{x}\sim\text{data}}\left[\ln p(\boldsymbol{x})\right] \qquad p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})}/Z$$

Learn



$$\nabla_\theta \mathscr{L} = \langle \nabla_\theta E \rangle_{\text{data}} - \langle \nabla_\theta E \rangle_{\text{model}}$$

2210.10318

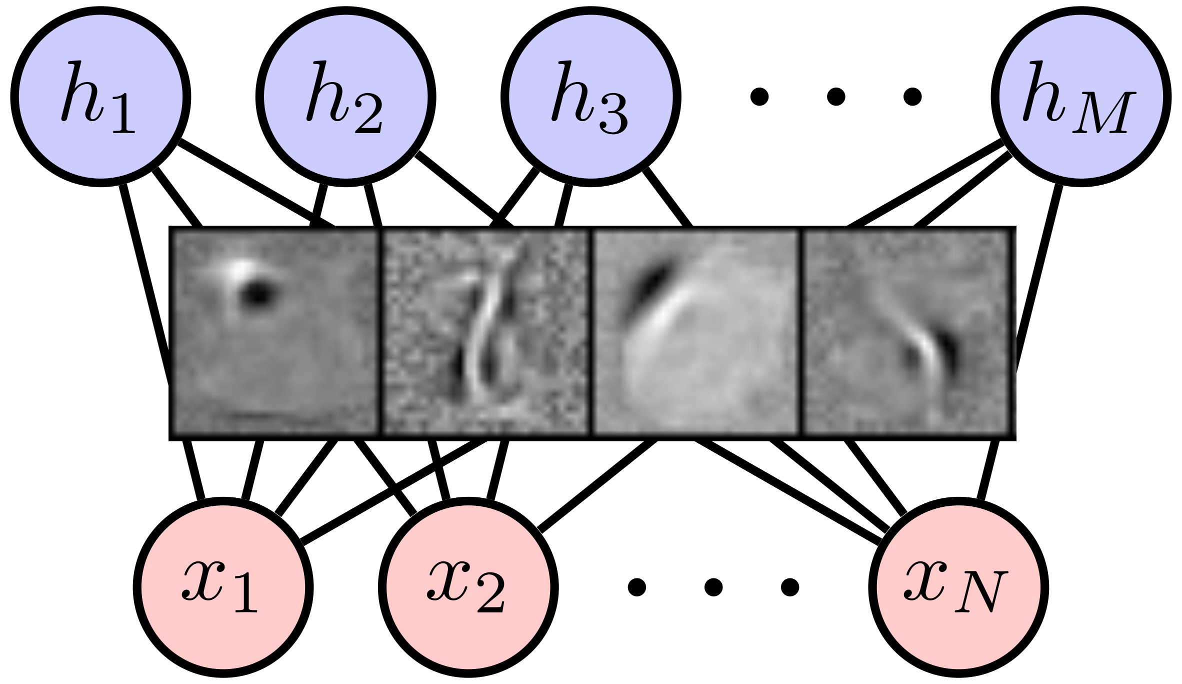
GAUSSIAN-BERNOULLI RBMS WITHOUT TEARS

Renjie Liao*[1], Simon Kornblith[2], Mengye Ren[3], David J. Fleet[2,4,5], Geoffrey Hinton[2,4,5]

Boltzmann machines

$$\mathscr{L} = -\,\mathbb{E}_{\boldsymbol{x}\sim\text{data}}\left[\ln p(\boldsymbol{x})\right] \qquad p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})}/Z$$

Learn



Generate

$$\nabla_\theta \mathscr{L} = \langle \nabla_\theta E \rangle_{\text{data}} - \langle \nabla_\theta E \rangle_{\text{model}}$$

2210.10318

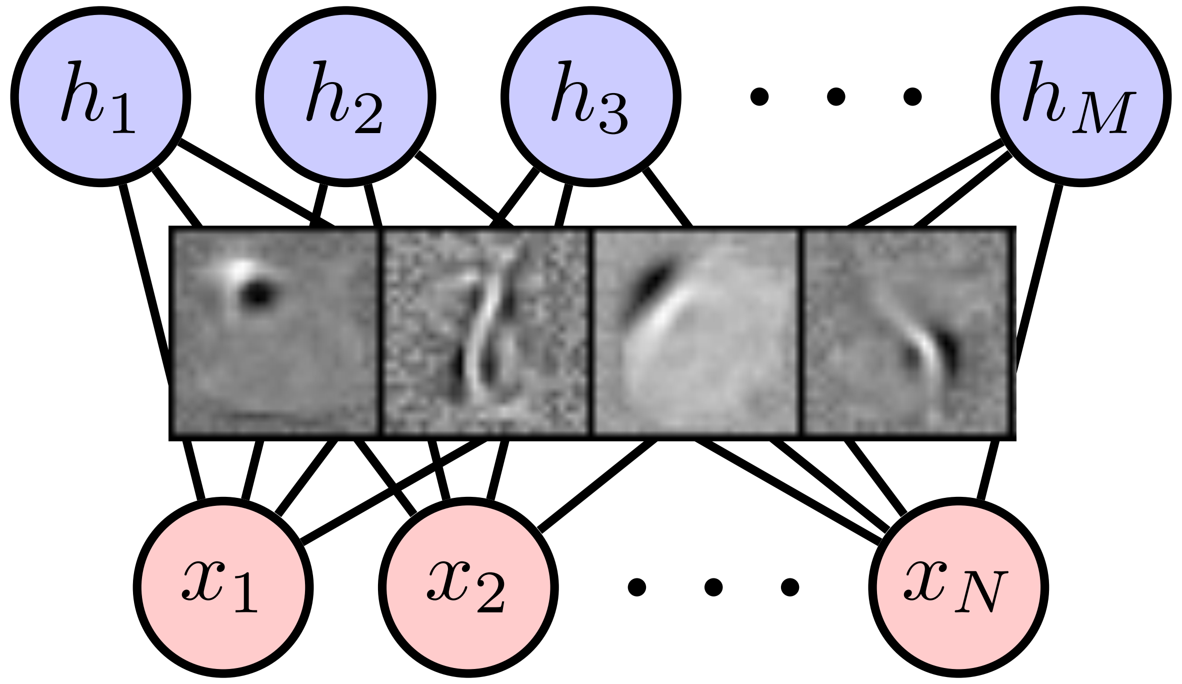
GAUSSIAN-BERNOULLI RBMS WITHOUT TEARS

Renjie Liao*[1], Simon Kornblith[2], Mengye Ren[3], David J. Fleet[2,4,5], Geoffrey Hinton[2,4,5]

Boltzmann machines

$$\mathscr{L} = -\,\mathbb{E}_{\boldsymbol{x}\sim\text{data}}\left[\ln p(\boldsymbol{x})\right] \qquad p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})}/Z$$

Learn



Generate

$$\nabla_\theta \mathscr{L} = \langle \nabla_\theta E \rangle_{\text{data}} - \langle \nabla_\theta E \rangle_{\text{model}}$$

2210.10318

GAUSSIAN-BERNOULLI RBMS WITHOUT TEARS

Renjie Liao*[1], Simon Kornblith[2], Mengye Ren[3], David J. Fleet[2,4,5], Geoffrey Hinton[2,4,5]

Diffusion models

I will follow the score-matching route https://yang-song.net/blog/2021/score/

Score matching

Minimizing Fisher divergence avoids the intractable partition function problem

$$\underset{\substack{\uparrow \\ \text{target}}}{\mathbb{F}(\pi} \parallel \underset{\substack{\uparrow \\ \text{model}}}{p)} = \int d\boldsymbol{x}\, \pi(\boldsymbol{x}) \left| \nabla_{\boldsymbol{x}} \ln \pi(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \ln p(\boldsymbol{x}) \right|^2$$

However, it brings up another problem

How to learn the model without knowing $\nabla \ln \pi$?

Implicit score matching

Integrate by parts   Hyvarinen JMLR '05

$$\mathbb{F}(\pi \parallel p) = \int d\boldsymbol{x}\, \pi(\boldsymbol{x}) \Big(|\nabla \ln p(\boldsymbol{x})|^2 + 2\nabla^2 \ln p(\boldsymbol{x}) \Big) + \text{const}\,.$$

The laplacian term can be difficult to compute

Cheaper stochastic estimate:
Song et al, 1905.07088

Curiously, the same expression for the kinetic energy of a wavefunction

$$\frac{\nabla^2 \psi}{\psi} = |\nabla \ln \psi|^2 + \nabla^2 \ln \psi$$

Forward laplacian:
Li et al, 2307.08214

Denoising score matching

Perturb data with small noise Vincent 2011

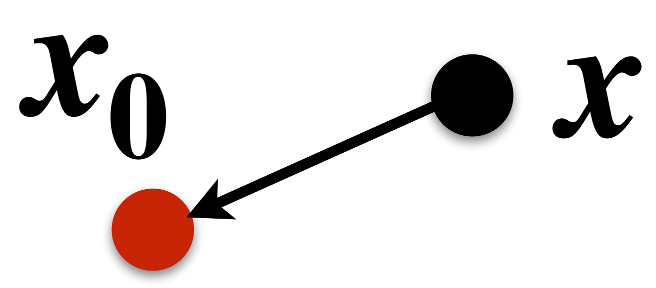
$$q(\boldsymbol{x}) = \int q(\boldsymbol{x} \,|\, \boldsymbol{x}_0)\pi(\boldsymbol{x}_0)d\boldsymbol{x}_0 \qquad q(\boldsymbol{x} \,|\, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{x}_0, \sigma^2)$$

Fisher divergence between perturbed data and model is computable

$$\mathbb{F}(q \,\|\, p) = \mathbb{E}_{\boldsymbol{x} \sim q(\boldsymbol{x})} |\nabla \ln q(\boldsymbol{x}) - \nabla \ln p(\boldsymbol{x})|^2$$

$$= \mathbb{E}_{\boldsymbol{x}_0 \sim \pi(\boldsymbol{x}_0)} \mathbb{E}_{\boldsymbol{x} \sim q(\boldsymbol{x}|\boldsymbol{x}_0)} |\nabla \ln q(\boldsymbol{x} \,|\, \boldsymbol{x}_0) - \nabla \ln p(\boldsymbol{x})|^2 + \text{const}\,.$$

$$\frac{\boldsymbol{x}_0 - \boldsymbol{x}}{\sigma^2}$$ the restoring force



Claim:

$$\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}|\nabla \ln q(\boldsymbol{x}) - \boldsymbol{s}_\theta|^2 = \mathbb{E}_{\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0)}\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0)}|\nabla \ln q(\boldsymbol{x}|\boldsymbol{x}_0) - \boldsymbol{s}_\theta|^2 + \underset{\substack{\downarrow \\ \text{Independent} \\ \text{of } \theta}}{\text{const}}\,.$$

$$\text{score} \quad \boldsymbol{s}_\theta = \nabla \ln p_\theta(\boldsymbol{x})$$

Proof:

$$\mathbb{E}_{x_0\sim\pi(x_0)}\mathbb{E}_{x\sim q(x|x_0)}|s|^2 = \int dx_0 \int dx \pi(x_0) q(x|x_0)|s|^2 = \int dx q(x)|s|^2 = \mathbb{E}_{x\sim q(x)}|s^2|$$

$$\mathbb{E}_{x_0\sim\pi(x_0)}\mathbb{E}_{x\sim q(x|x_0)}[s\cdot\nabla \ln q(x|x_0)] = \int dx_0 \int dx \pi(x_0) q(x|x_0)\frac{s\cdot\nabla q(x|x_0)}{q(x|x_0)}$$

$$= \int dx_0 \int dx \pi(x_0) s\cdot\nabla q(x|x_0)$$

$$= \int dx s\cdot\nabla q(x) = \mathbb{E}_{x\sim q(x)}[s\cdot\nabla \ln q(x)]$$

Why score matching did not take off?

Hard to sample between modes with Langevin dynamics

$$\boldsymbol{x}_{t+\epsilon} = \boldsymbol{x}_t + \frac{\epsilon}{2}\nabla \ln p(\boldsymbol{x}_t) + \sqrt{\epsilon}\mathcal{N}(0,I) \qquad \boldsymbol{x}_t \sim p(\boldsymbol{x}_t)$$

for $\epsilon \to 0, \quad t \to \infty$

From denoising score matching to diffusion model

Song et al, Generative modeling by estimating gradients of the data distribution, 1907.05600

Built upon this intuition, we propose to improve score-based generative modeling by 1) *perturbing the data using various levels of noise*; and 2) *simultaneously estimating scores corresponding to all noise levels by training a single conditional score network.* After training, when using Langevin dynamics to generate samples, we initially use scores corresponding to large noise, and gradually anneal down the noise level. This helps smoothly transfer the benefits of large noise levels to low noise levels where the perturbed data are almost indistinguishable from the original ones. In what follows, we will elaborate more on the details of our method, including the architecture of our score networks, the training objective, and the annealing schedule for Langevin dynamics.

Sohl-Dickstein et al, Deep unsupervised learning using nonequilibrium thermodynamics, 1503.03585

The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This ap-

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

Ho et al, Denoising diffusion probabilistic models, 2006.11239

From denoising score matching to diffusion model

The objective of denoising diffusion probabilistic model

Song et al, 1907.05600
Ho et al, 2006.11239

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

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

Sample with annealed Langevin dynamics with decreasing steps ϵ_t

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \frac{\epsilon_t}{2} \boldsymbol{s}(\boldsymbol{x}_t, t) + \sqrt{\epsilon_t} \mathcal{N}(0, I)$$

A tale of three equations

Langevin equation (SDE)

$$\boldsymbol{x}_{t+dt} = \boldsymbol{x}_t + \boldsymbol{f}dt + \sqrt{2dt}\,\mathcal{N}(0,I)$$

Fokker-Planck equation (PDE)

$$\frac{\partial p(\boldsymbol{x},t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x},t)\boldsymbol{f}\right] - \nabla^2 p(\boldsymbol{x},t) = 0$$

"Particle method" (ODE)

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f} - \nabla \ln p(\boldsymbol{x},t) \equiv \boldsymbol{v}$$

(Another way to reverse the diffusion is via the reverse-time SDE Anderson 1982)

Maoutsa et al, 2006.00702
Song et al, 2011.13456

A tale of three equations

Langevin equation (SDE)

$$\boldsymbol{x}_{t+dt} = \boldsymbol{x}_t + \boldsymbol{f}dt + \sqrt{2dt}\,\mathcal{N}(0,I)$$

Fokker-Planck equation (PDE)

$$\frac{\partial p(\boldsymbol{x},t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x},t)\left(\boldsymbol{f} - \nabla \ln p(\boldsymbol{x},t)\right)\right] = 0$$

"Particle method" (ODE)

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f} - \nabla \ln p(\boldsymbol{x},t) \equiv \boldsymbol{v}$$

(Another way to reverse the diffusion is via the reverse-time SDE Anderson 1982)

Maoutsa et al, 2006.00702
Song et al, 2011.13456

from Langevin to Fokker-Planck

$$\mathcal{P}(\vec{x},t)=\int \mathrm{d}^3\vec{x}\,'\left(\frac{1}{4\pi D\epsilon}\right)^{3/2}\exp\left[-\frac{\left(\vec{x}-\vec{x}\,'-\epsilon\vec{v}(\vec{x}\,')\right)^2}{4D\epsilon}\right]\mathcal{P}(\vec{x}\,',t-\epsilon), \tag{9.18}$$

and simplified by the change of variables,

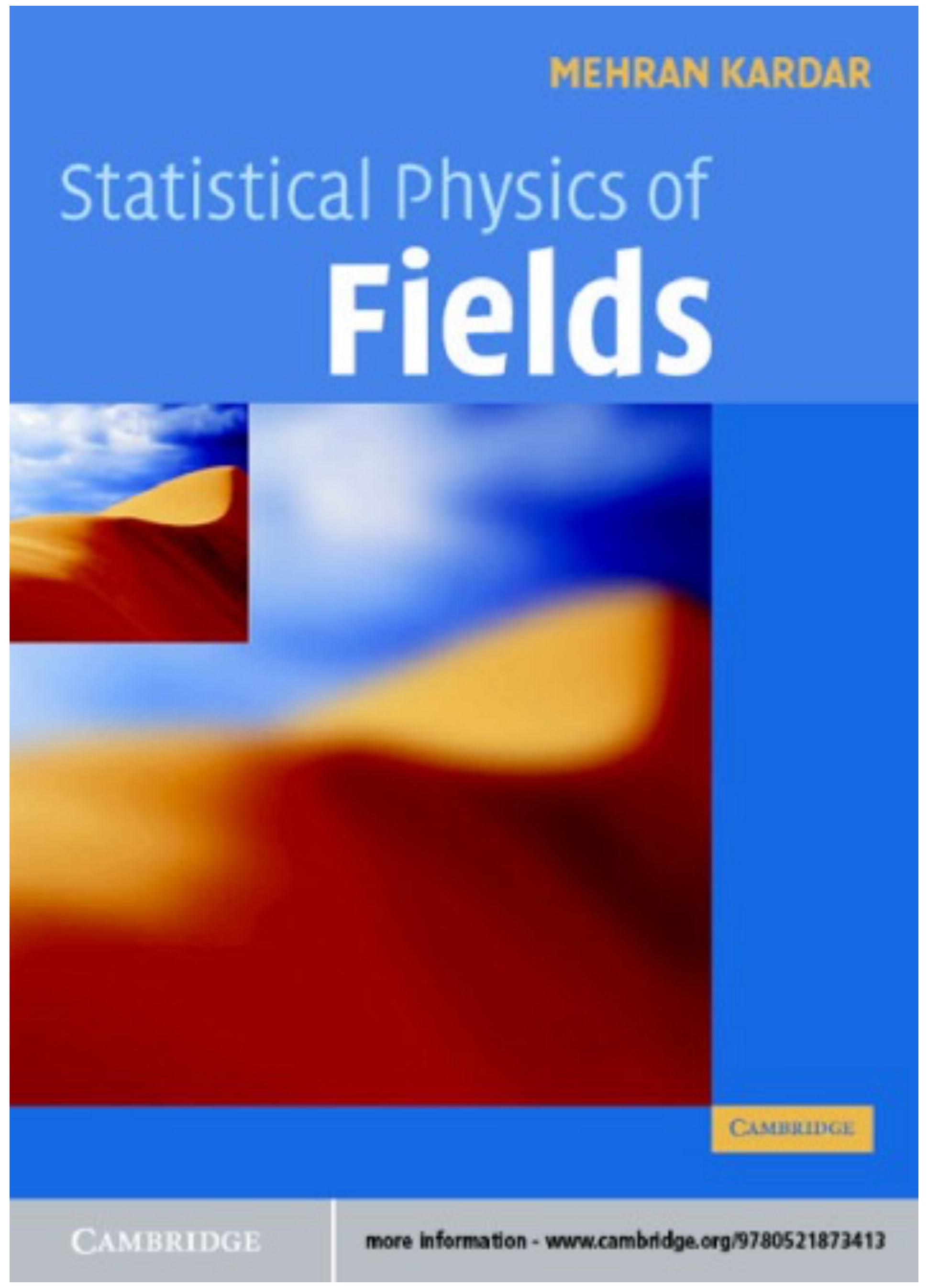
$$\begin{aligned}\vec{y}&=\vec{x}\,'+\epsilon\vec{v}(\vec{x}\,')-\vec{x}\quad\Longrightarrow\\ \mathrm{d}^3\vec{y}&=\mathrm{d}^3\vec{x}\,'\left(1+\epsilon\nabla\cdot\vec{v}(\vec{x}')\right)=\mathrm{d}^3\vec{x}\,'\left(1+\epsilon\nabla\cdot\vec{v}(\vec{x})+\mathcal{O}(\epsilon^2)\right).\end{aligned} \tag{9.19}$$

Keeping only terms at order of ϵ, we obtain

$$\begin{aligned}\mathcal{P}(\vec{x},t)&=[1-\epsilon\nabla\cdot\vec{v}(\vec{x})]\int \mathrm{d}^3\vec{y}\left(\frac{1}{4\pi D\epsilon}\right)^{3/2}\mathrm{e}^{-\frac{y^2}{4D\epsilon}}\ \mathcal{P}(\vec{x}+\vec{y}-\epsilon\vec{v}(\vec{x}),t-\epsilon)\\ &=[1-\epsilon\nabla\cdot\vec{v}(\vec{x})]\int \mathrm{d}^3\vec{y}\left(\frac{1}{4\pi D\epsilon}\right)^{3/2}\mathrm{e}^{-\frac{y^2}{4D\epsilon}}\\ &\times\left[\mathcal{P}(\vec{x},t)+(\vec{y}-\epsilon\vec{v}(\vec{x}))\cdot\nabla\mathcal{P}+\frac{y_iy_j-2\epsilon y_iv_j+\epsilon^2v_iv_j}{2}\nabla_i\nabla_j\mathcal{P}-\epsilon\frac{\partial\mathcal{P}}{\partial t}+\mathcal{O}(\epsilon^2)\right]\\ &=[1-\epsilon\nabla\cdot\vec{v}(\vec{x})]\left[\mathcal{P}-\epsilon\vec{v}\cdot\nabla+\epsilon D\nabla^2\mathcal{P}-\epsilon\frac{\partial\mathcal{P}}{\partial t}+\mathcal{O}(\epsilon^2)\right].\end{aligned} \tag{9.20}$$

Equating terms at order of ϵ leads to the *Fokker–Planck equation*,

$$\frac{\partial\mathcal{P}}{\partial t}+\nabla\cdot\vec{J}=0,\quad\text{with}\quad\vec{J}=\vec{v}\mathcal{P}-D\nabla\mathcal{P}. \tag{9.21}$$



Lessons from diffusion models

Continuous normalizing flow has great potential: diffusion model is an "existence proof"

Going beyond maximum likelihood estimation training (even if we can)

Break the loss into small pieces, sample them (layer-wise regression)

https://blog.alexalemi.com/diffusion.html

The conditional trick (originated from denoising score matching Vincent 2011)

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

Lessons from diffusion models

Continuous normalizing flow has great potential: diffusion model is an "existence proof"

Going beyond maximum likelihood estimation training (even if we can)

Break the loss into small pieces, sample them (layer-wise regression)

https://blog.alexalemi.com/diffusion.html

The conditional trick (originated from denoising score matching Vincent 2011)

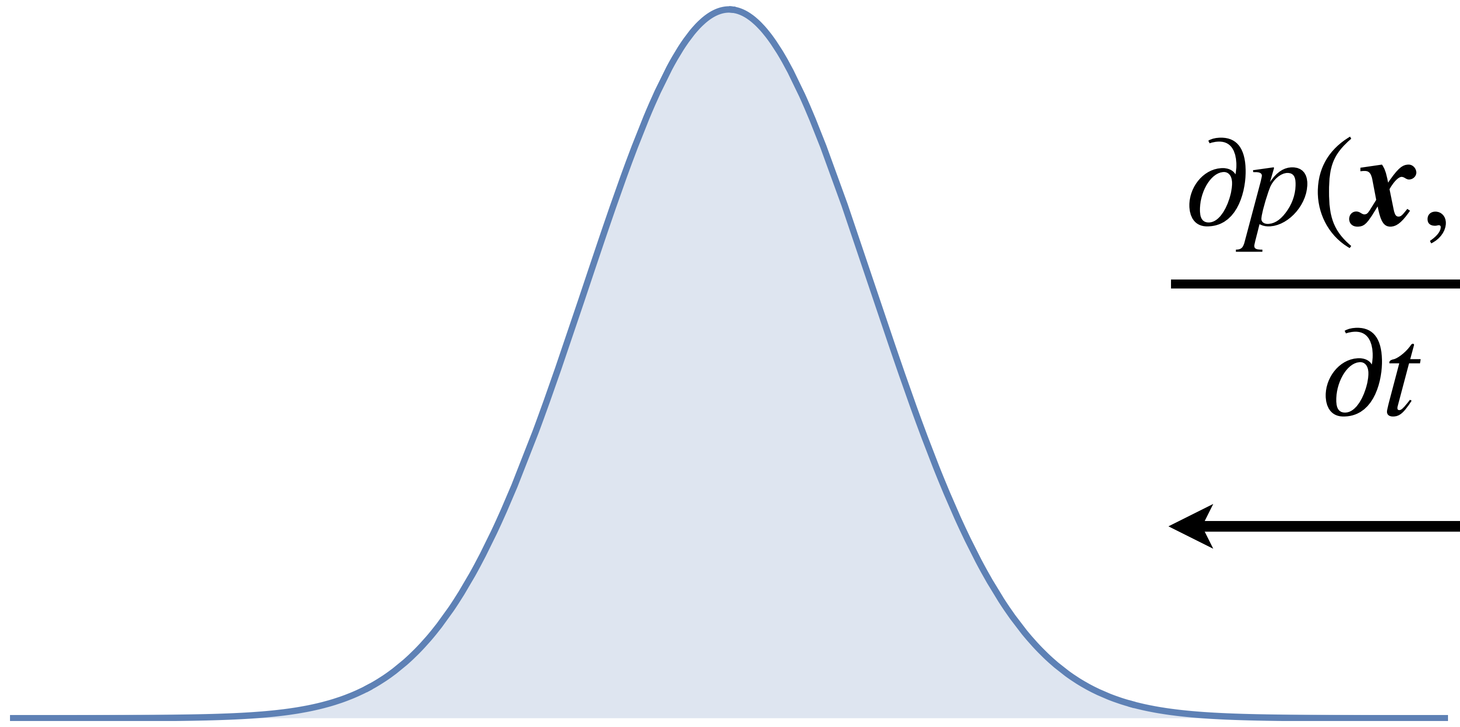
$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{t\sim\mathcal{U}(0,T)}}_{\text{diffusion time } t} \underbrace{\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)}}_{\text{diffused data } \mathbf{x}_t} ||\underbrace{\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}_{\text{neural network}} - \underbrace{\nabla_{\mathbf{x}_t}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

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

Flow matching

$p(\boldsymbol{x},1) = \mathcal{N}(0,I)$

base distribution



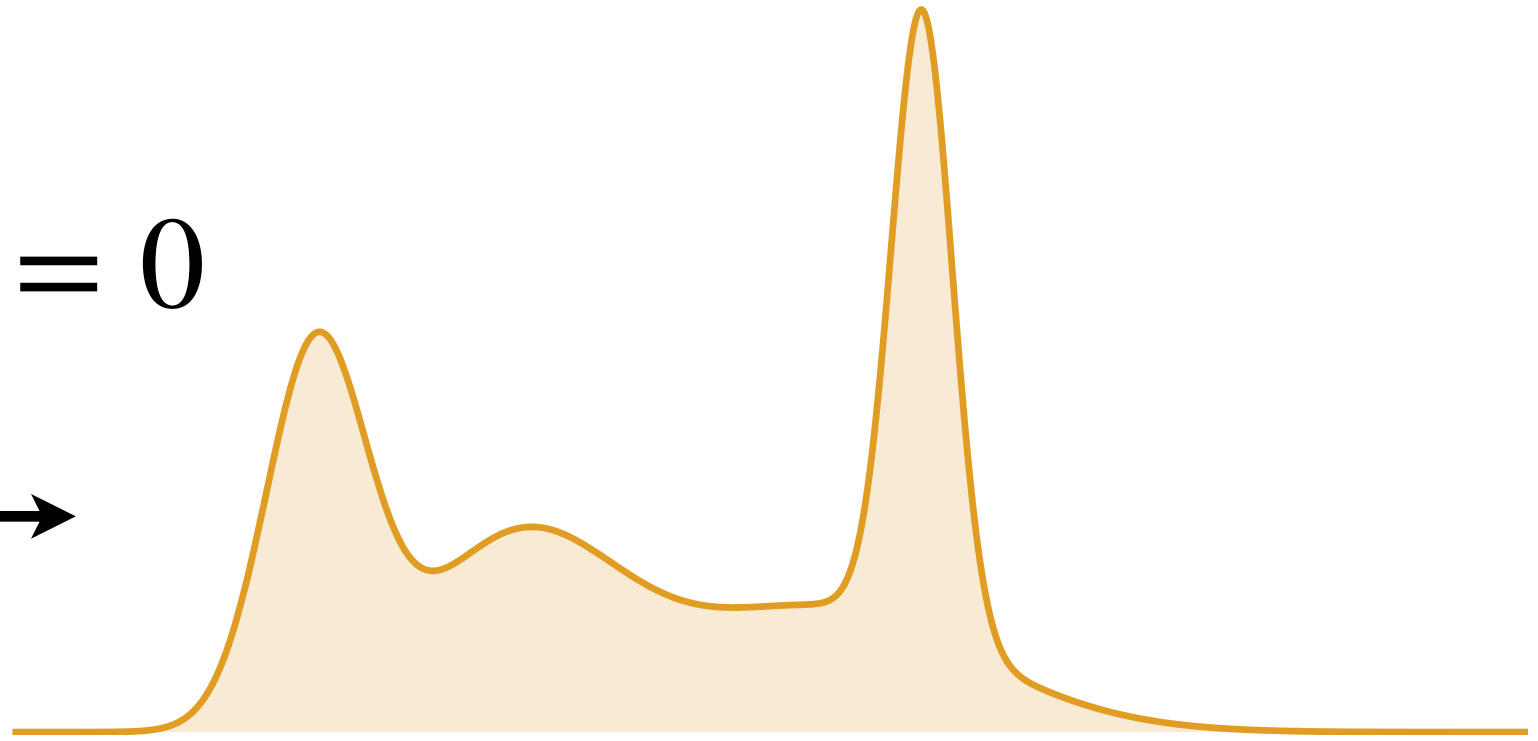
ground truth

velocity field

$$\frac{\partial p(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t)\right] = 0$$

$p(\boldsymbol{x},0) = q(\boldsymbol{x})$

data distribution



$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)} \left|\boldsymbol{v}_\theta(\boldsymbol{x}, t) - \boldsymbol{u}(\boldsymbol{x}, t)\right|^2$$

Liu et al 2209.03003, Albergo et al, 2209.15571, Lipman et al, 2210.02747

The "conditional" trick

Given a conditional continuity equation

$$\frac{\partial p(\boldsymbol{x} \,|\, \boldsymbol{x}_0, t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x} \,|\, \boldsymbol{x}_0, t)\boldsymbol{u}(\boldsymbol{x} \,|\, \boldsymbol{x}_0, t) \right] = 0$$

Then, up to a constant, we have

$$\mathscr{L} = \mathbb{E}_{t \sim \mathscr{U}(0,1)} \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)} \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{x}_0, t)} \left| \boldsymbol{v}_\theta(\boldsymbol{x}, t) - \boldsymbol{u}(\boldsymbol{x} \,|\, \boldsymbol{x}_0, t) \right|^2$$

We can learn the ground truth velocity by regressing on the conditional velocity

Claim: $\mathscr{L}_{\mathrm{CFM}} = \mathscr{L}_{\mathrm{FM}} + \mathrm{const}\,.$

where $\mathscr{L}_{\mathrm{FM}} = \mathbb{E}_{t\sim\mathscr{U}(0,1)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)}\left|\boldsymbol{v}_\theta(\boldsymbol{x},t) - \boldsymbol{u}(\boldsymbol{x},t)\right|^2$

$$\mathscr{L}_{\mathrm{CFM}} = \mathbb{E}_{t\sim\mathscr{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left|\boldsymbol{v}_\theta(\boldsymbol{x},t) - \boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\right|^2$$

$$p(\boldsymbol{x},t) = \int p(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\,q(\boldsymbol{x}_0)d\boldsymbol{x}_0 \qquad p(\boldsymbol{x},t)\boldsymbol{u}(\boldsymbol{x},t) = \int p(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\,q(\boldsymbol{x}_0)d\boldsymbol{x}_0$$

Proof:

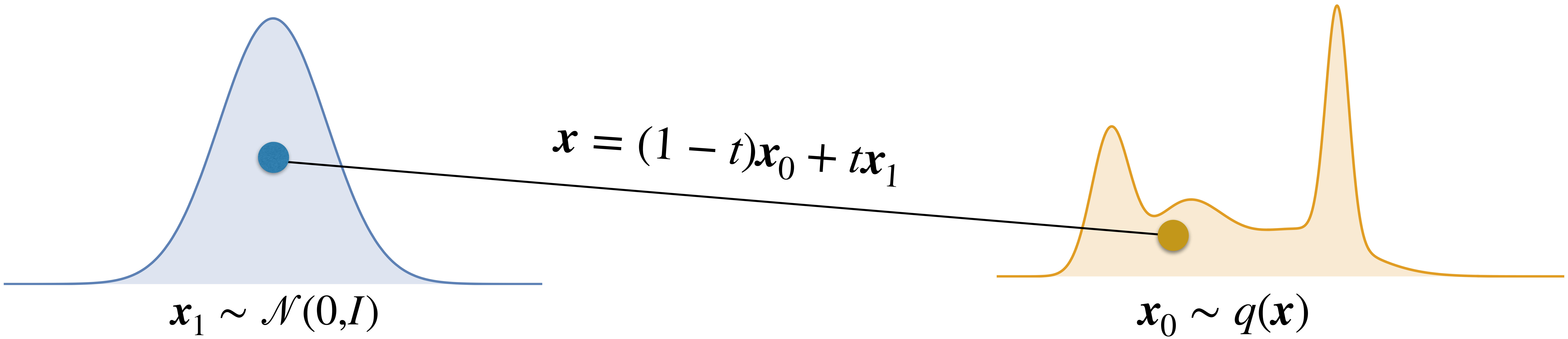
$$\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left|\boldsymbol{v}_\theta\right|^2 = \int d\boldsymbol{x}_0\int d\boldsymbol{x}q(\boldsymbol{x}_0)p(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\,|\,\boldsymbol{v}_\theta|^2 = \int d\boldsymbol{x}p(\boldsymbol{x},t)\,|\,\boldsymbol{v}_\theta|^2 = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)}\left|\boldsymbol{v}_\theta\right|^2$$

$$\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\boldsymbol{v}_\theta\cdot\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\right] = \int d\boldsymbol{x}_0\int d\boldsymbol{x}q(\boldsymbol{x}_0)p(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\left[\boldsymbol{v}_\theta\cdot\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_0,t)\right]$$

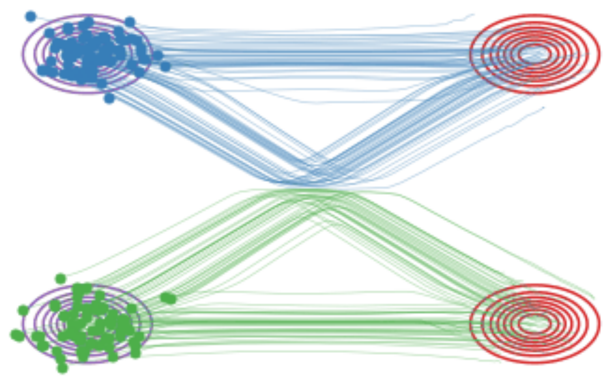
$$= \int d\boldsymbol{x}p(\boldsymbol{x},t)\boldsymbol{v}_\theta\cdot\boldsymbol{u}(\boldsymbol{x},t) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)}\left[\boldsymbol{v}_\theta\cdot\boldsymbol{u}(\boldsymbol{x},t)\right]$$

Examples of flow matching

$$p(\boldsymbol{x}\,|\,\boldsymbol{x}_0, t) = \mathcal{N}\left((1-t)\boldsymbol{x}_0, t^2\right) \qquad \boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_0, t) = \frac{d\boldsymbol{x}}{dt} = \boldsymbol{x}_1 - \boldsymbol{x}_0$$



$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0)}\mathbb{E}_{\boldsymbol{x}_1\sim\mathcal{N}(0,I)}\left|\boldsymbol{v}_\theta(\boldsymbol{x}, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\right|^2$$

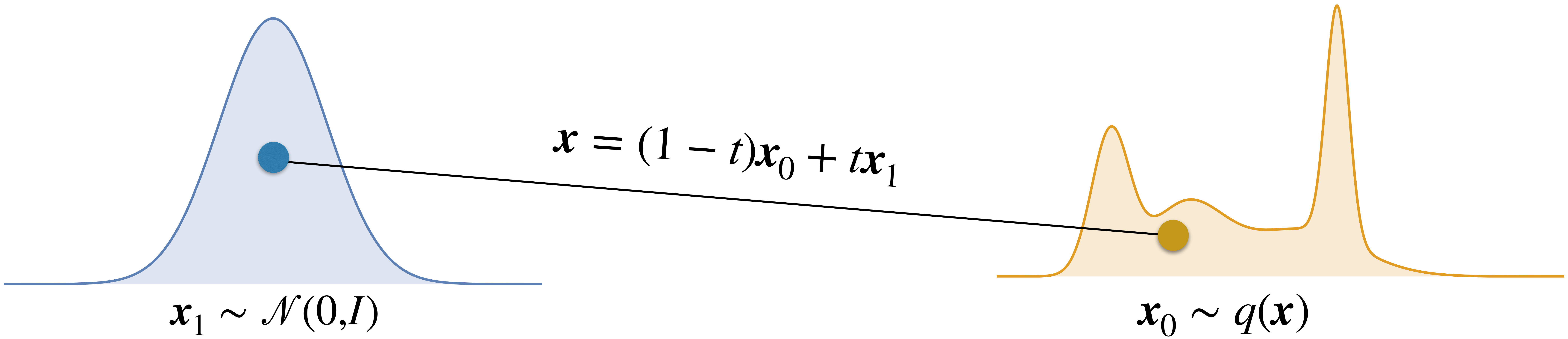


Causalizing linear interpolation with rectified flow 2209.03003

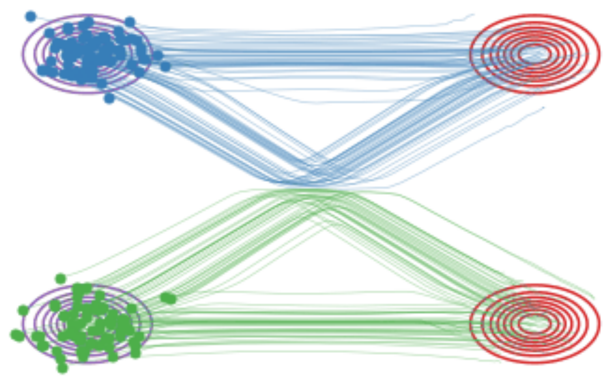
https://www.cs.utexas.edu/~lqiang/rectflow/html/intro.html

Examples of flow matching

$$p(\boldsymbol{x}\,|\,\boldsymbol{x}_0, t) = \mathcal{N}\left((1-t)\boldsymbol{x}_0, t^2\right) \qquad \boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_0, t) = \frac{d\boldsymbol{x}}{dt} = \boldsymbol{x}_1 - \boldsymbol{x}_0$$



$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0)}\mathbb{E}_{\boldsymbol{x}_1\sim\mathcal{N}(0,I)}\left|\boldsymbol{v}_\theta(\boldsymbol{x}, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\right|^2$$



Causalizing linear interpolation with rectified flow 2209.03003

https://www.cs.utexas.edu/~lqiang/rectflow/html/intro.html

Flow matching is all you need!

This framework contains various diffusion models as special cases

The base distribution does not have to be Gaussian

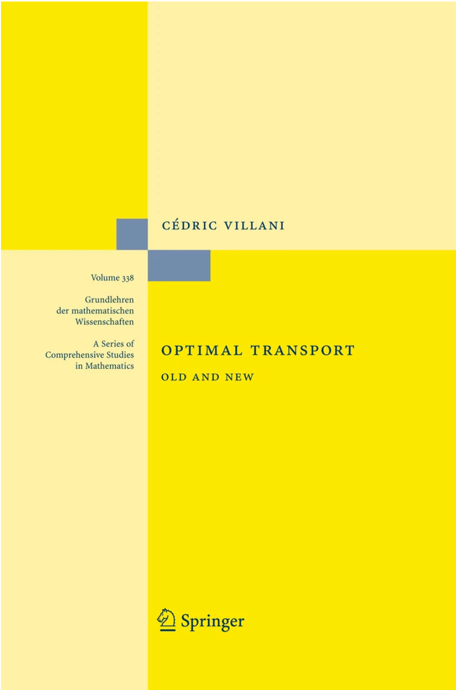
Fast generation with rectified transportation path (Liu et al 2209.03003)

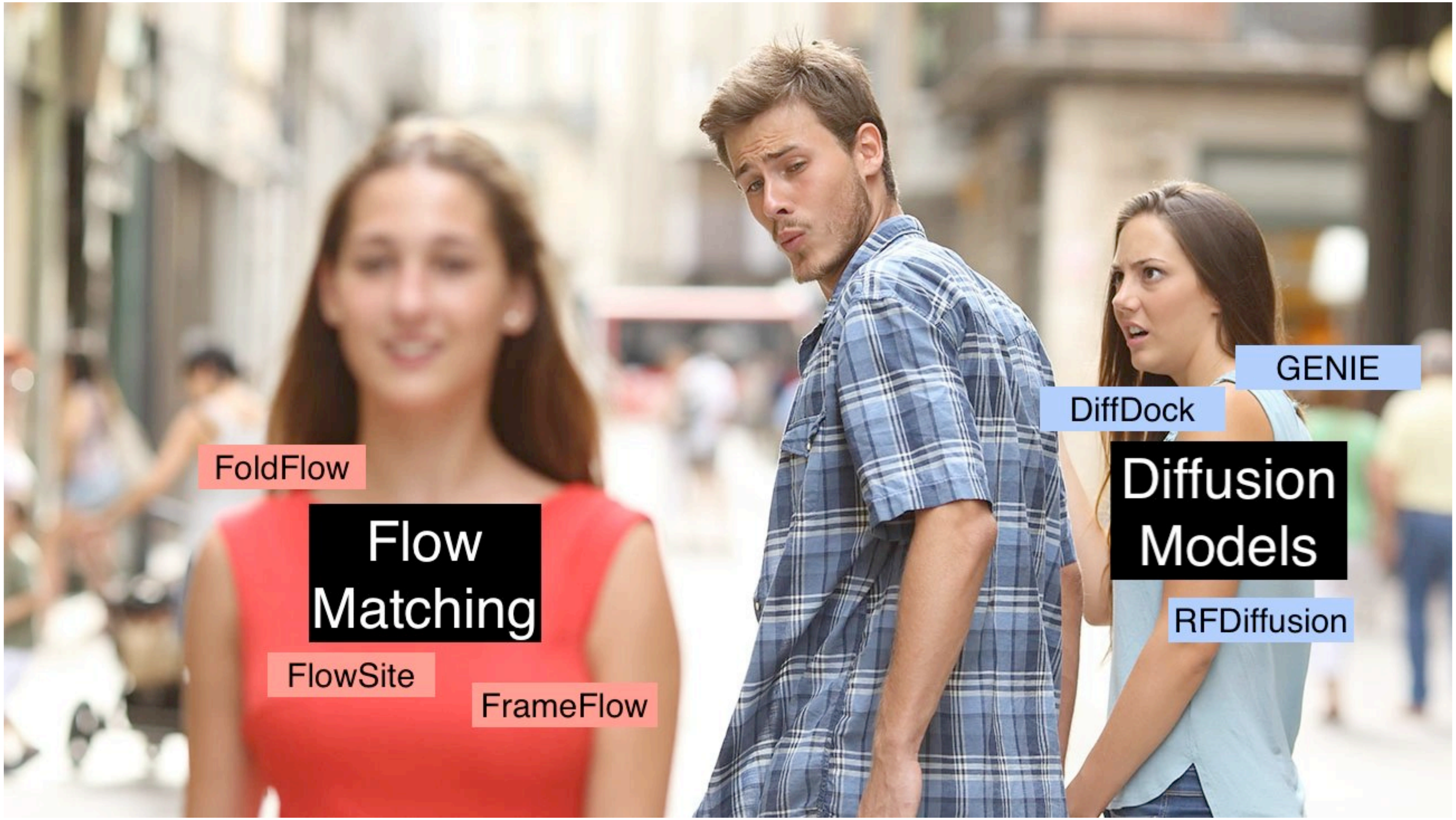
400x speedup compared to continuous normalizing flow (Albergo et al, 2209.15571)

Surpasses diffusion model on Imagenet in likelihood and sample quality (Lipman et al, 2210.02747)

Generalization to flow on Riemannian manifolds (Chen et al, 2302.03660)

Part II Optimal transport and Riemannian geometry



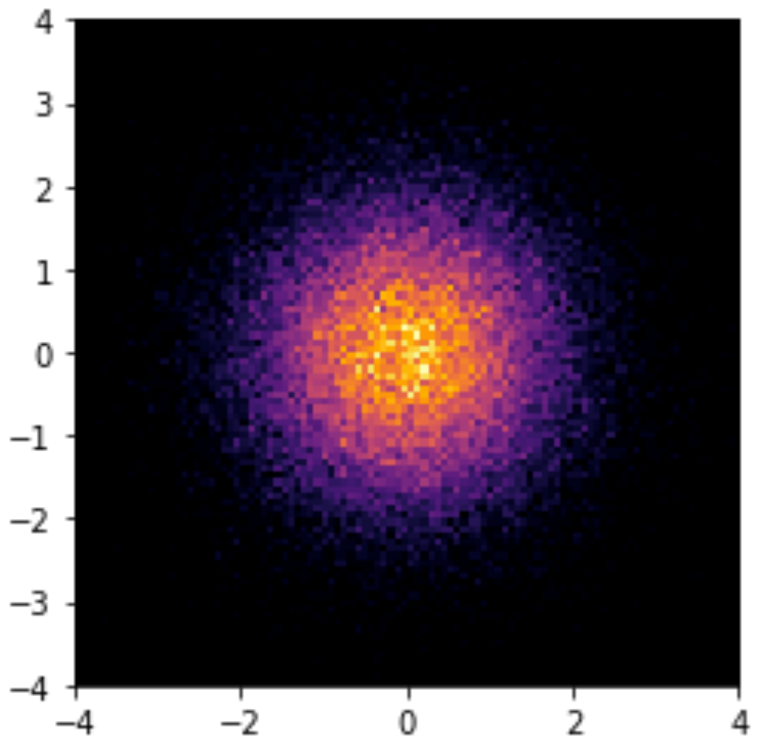


https://twitter.com/michael_galkin/status/1711845455817261409

Demo: free energy of classical Coulomb gas

$$\mathscr{L} = \mathbb{E}_{t\sim\mathscr{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_0\sim\mathscr{N}(0,I)}\mathbb{E}_{\boldsymbol{x}_1\sim\exp(-\beta E)/Z}\left|\boldsymbol{x}_1 - \boldsymbol{x}_0 - \boldsymbol{v}(\boldsymbol{x},t)\right|^2$$

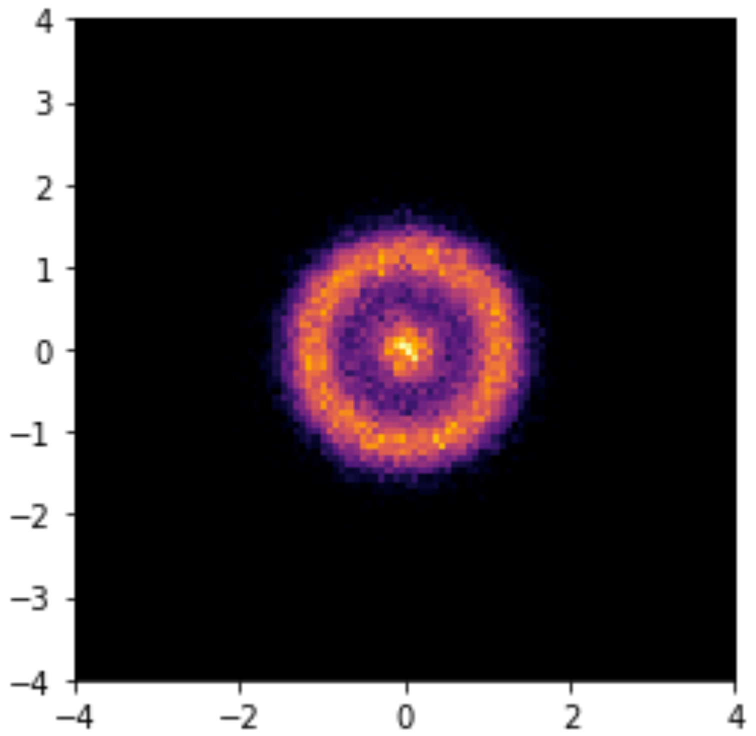
$$Z = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[e^{-\beta E(\boldsymbol{x}) - \ln q(\boldsymbol{x})}\right] \quad \ln q(\boldsymbol{x}) = \ln \mathscr{N}(0,I) - \int_0^1 \nabla\cdot\boldsymbol{v}dt$$



Base density

Gaussian samples

Interpolate samples to estimate free energy differences



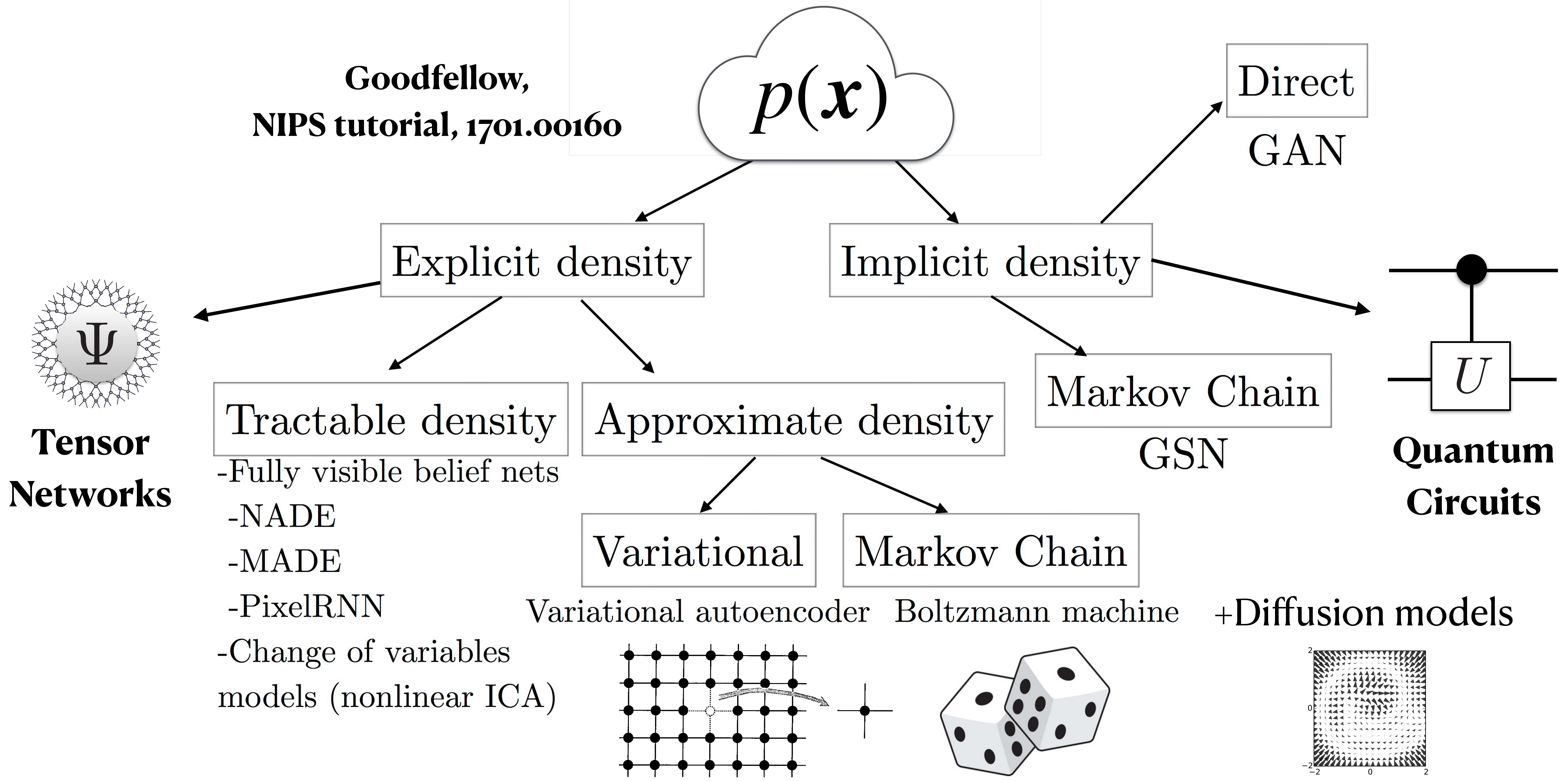
Target density

Monte Carlo samples

https://colab.research.google.com/drive/1t-Vk37Axxpo40B7uXFUNlk-zeCC2lcX3?usp=sharing

Jarzynski PRE '02, see also likelihood-based training of flows Wirnsberger et al, 2002.04913, 2111.08696

Generative models and their physics genes



Variational autoencoders

Kingma, Welling, 1312.6114

Close connection to the variational calculus we have learned

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

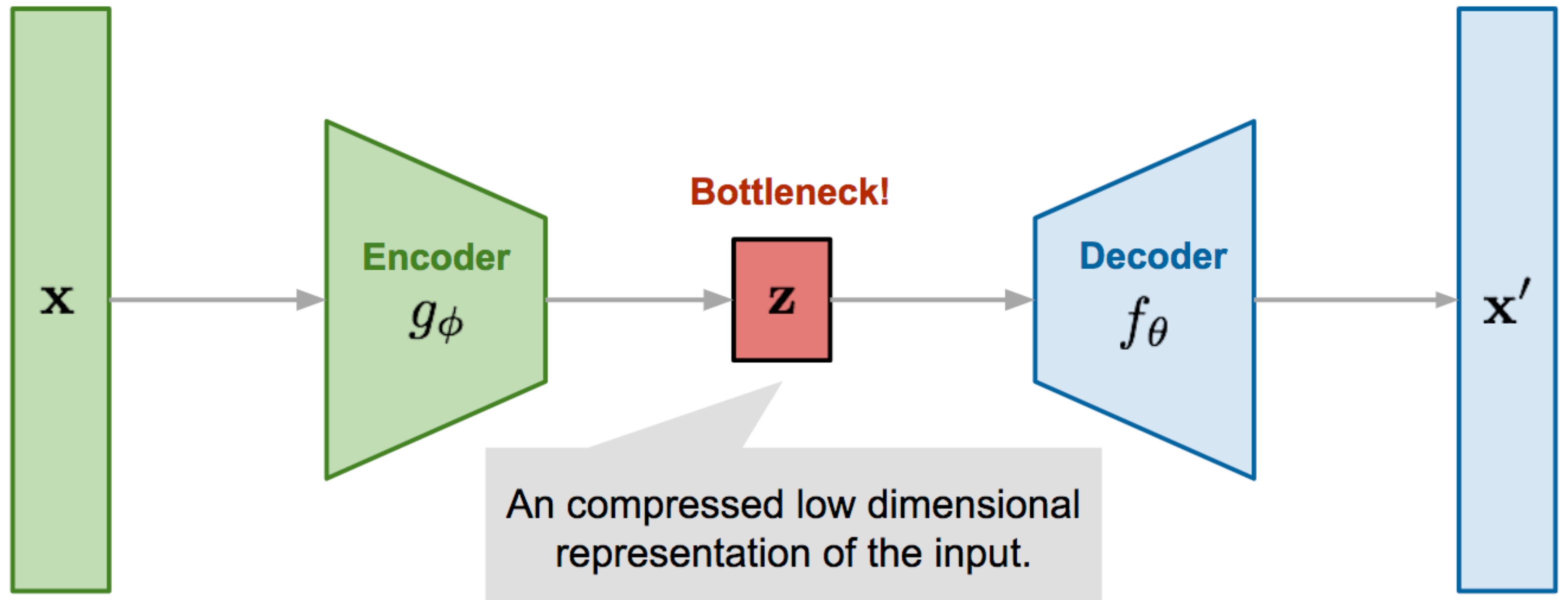
$$p(z\,|\,\boldsymbol{x}) = \frac{p(\boldsymbol{x}, z)}{p(\boldsymbol{x})}$$

Variational free energy

$$\int d\boldsymbol{x}\, q(\boldsymbol{x}) \left[\ln q(\boldsymbol{x}) + \beta E(\boldsymbol{x})\right] \geq -\ln Z$$

Variational Bayes/Variational inference

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



For each data we introduce

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

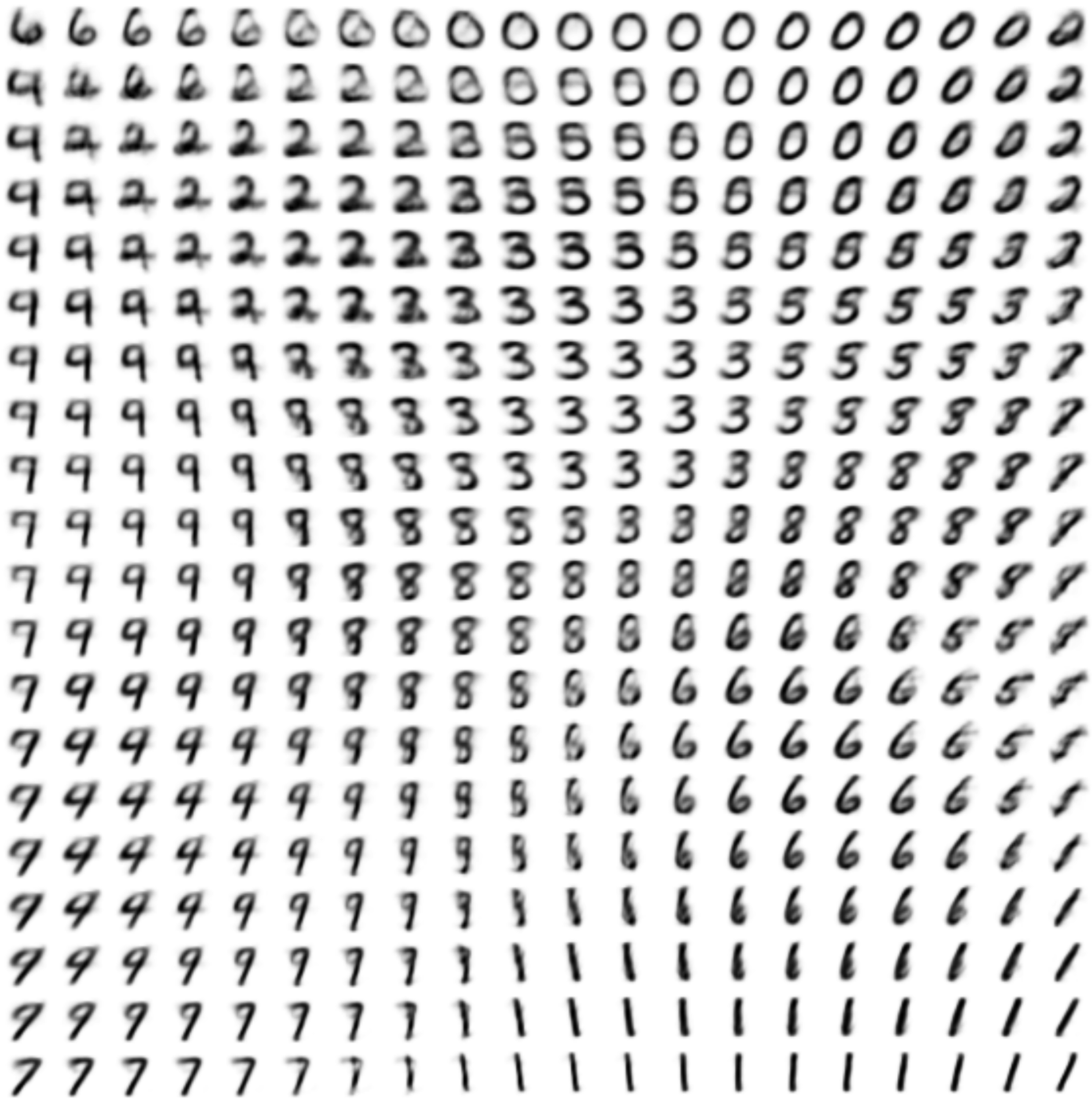
which is a variational upper bound of $-\ln p(\boldsymbol{x})$ since $\mathcal{L}(\boldsymbol{x}) + \ln p(\boldsymbol{x}) = \mathbb{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})) \geq 0$. We see that $q(\boldsymbol{z}|\boldsymbol{x})$ provides a variational approximation of the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

We can obtain an alternative form of the variational free energy

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

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 $\boldsymbol{\theta}, \boldsymbol{\phi}$ of the encoder and decoder.

http://wangleiphy.github.io/lectures/PILtutorial.pdf

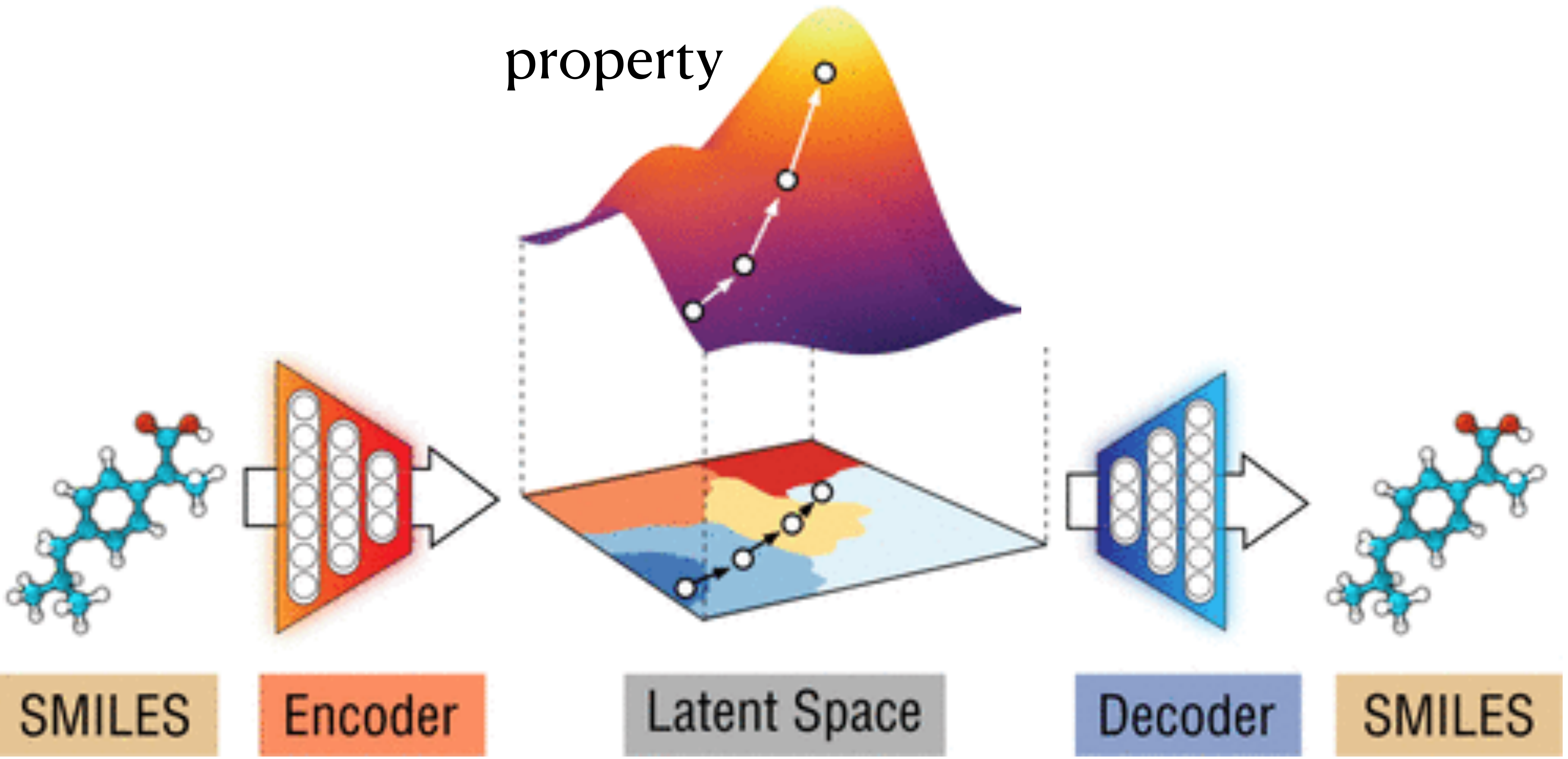


Learned MNIST latent space

Kingma, Welling, 1312.6114

Chemical design using continuous latent variables

Gomez-Bombarelli et al,1610.02415



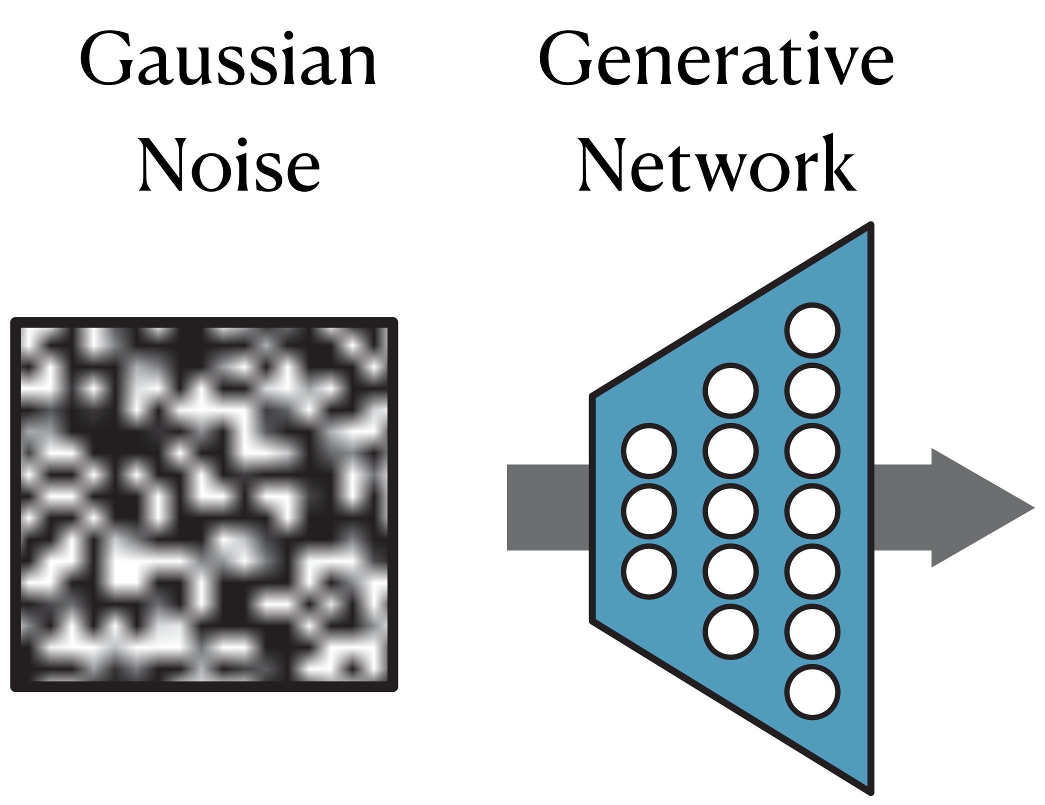
GAN

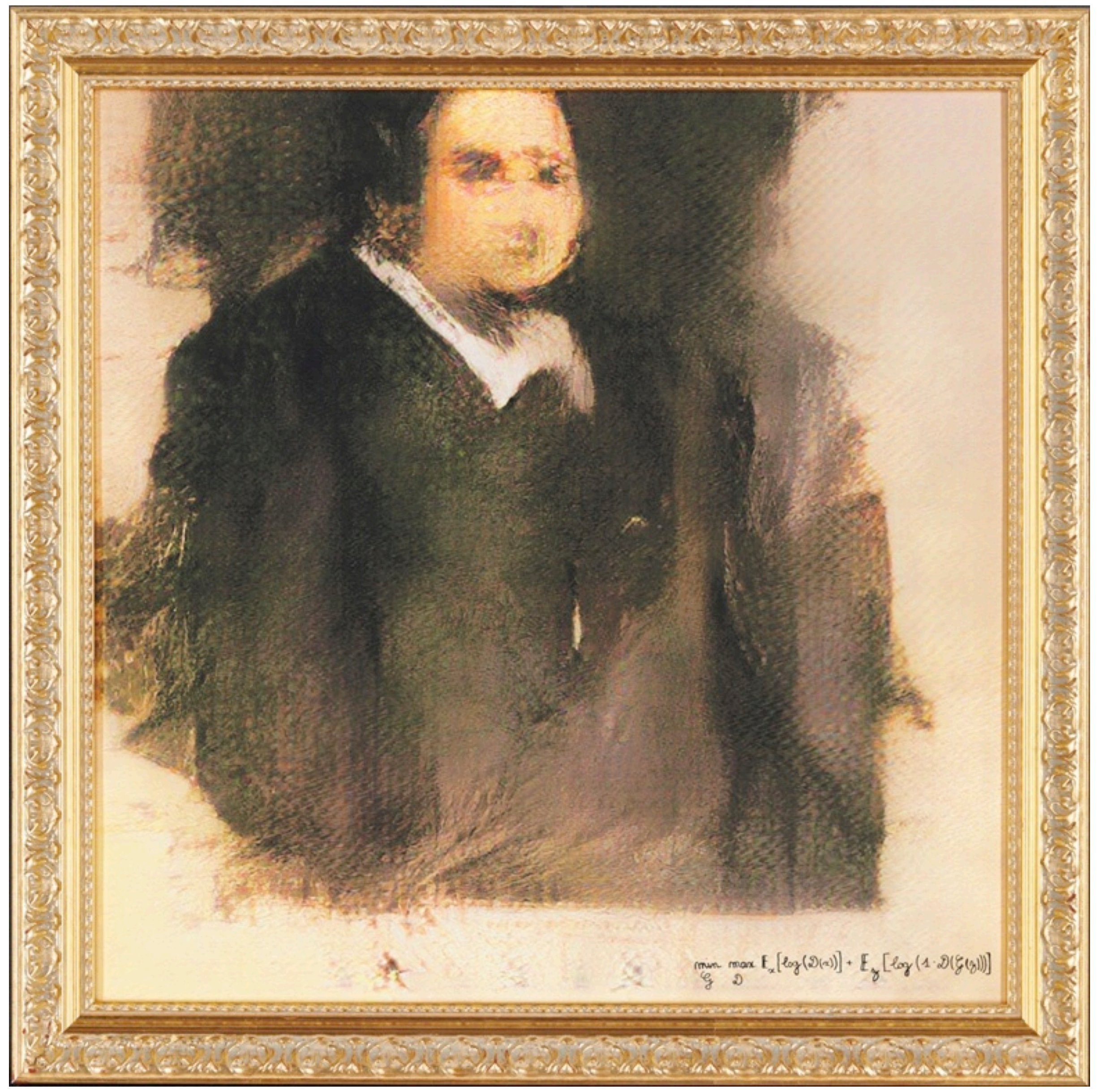
Likelihood free simulator

Prone to mode collapse “de”generate

More tricky to train than others

Performance have been surpassed by diffusion models

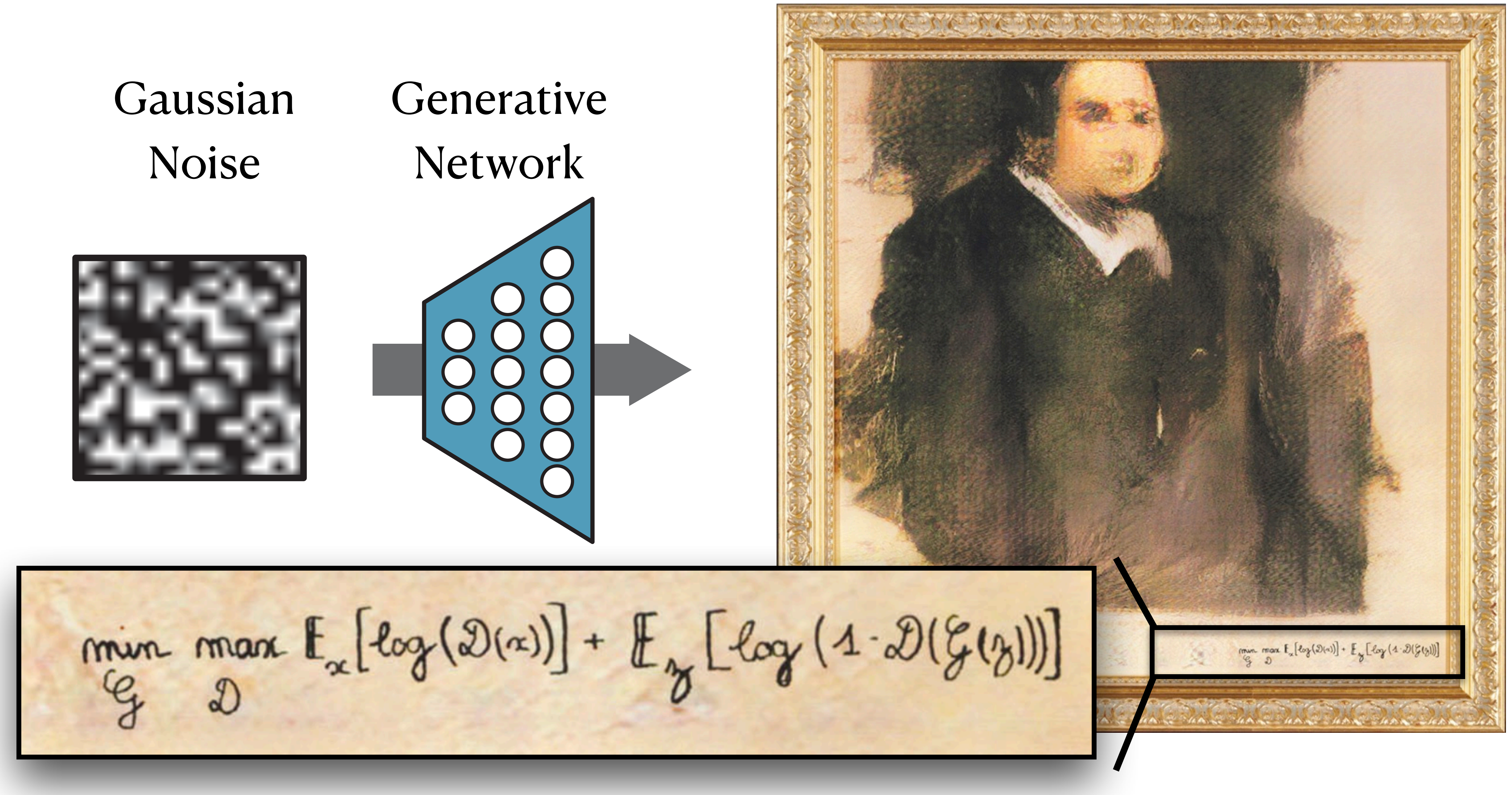




https://www.christies.com/Features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx

I found GAN to be less useful for quantitative scientific applications

GAN



https://www.christies.com/Features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx

Likelihood free simulator

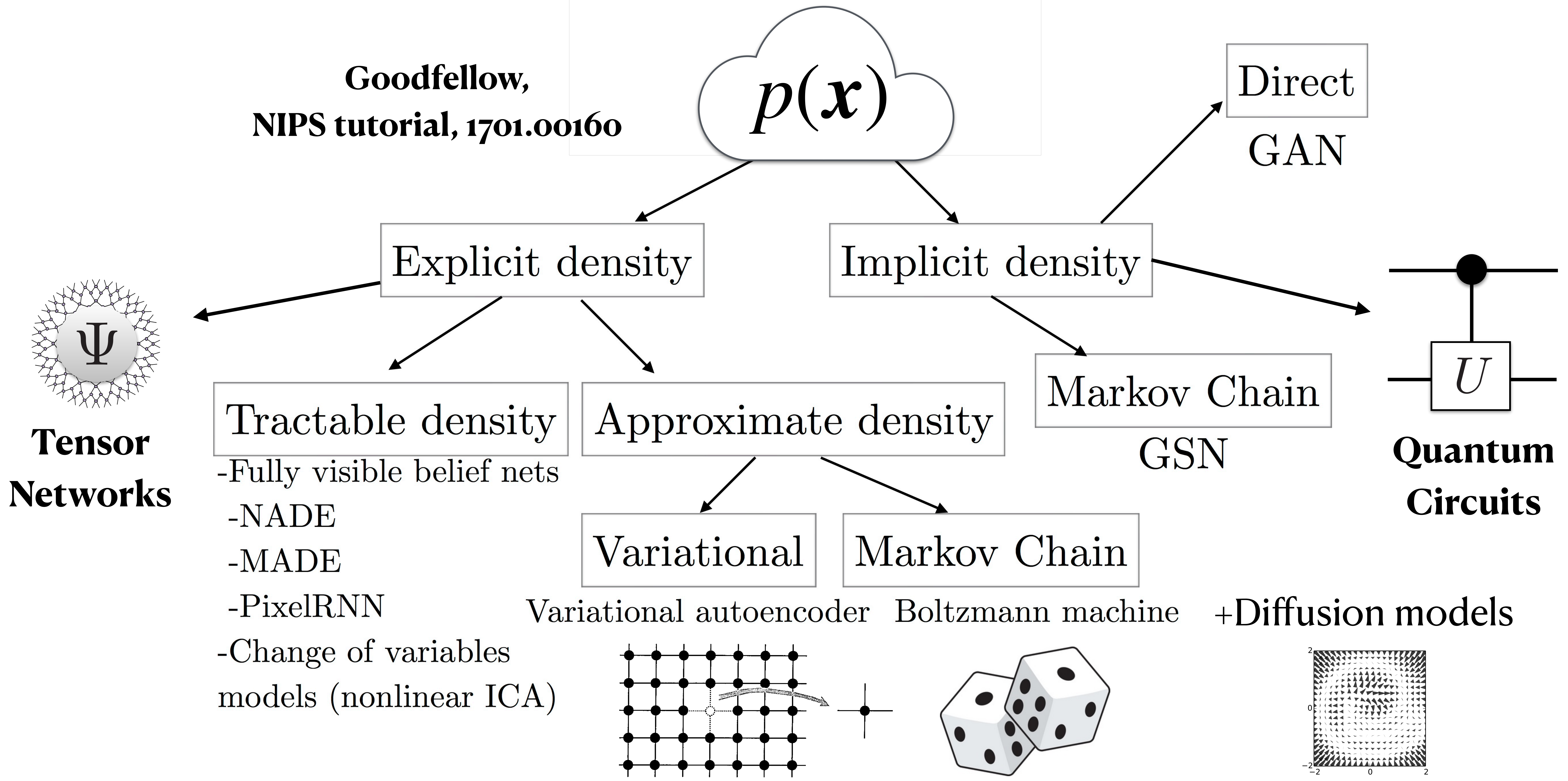
Prone to mode collapse “de”generate

More tricky to train than others

Performance have been surpassed by diffusion models

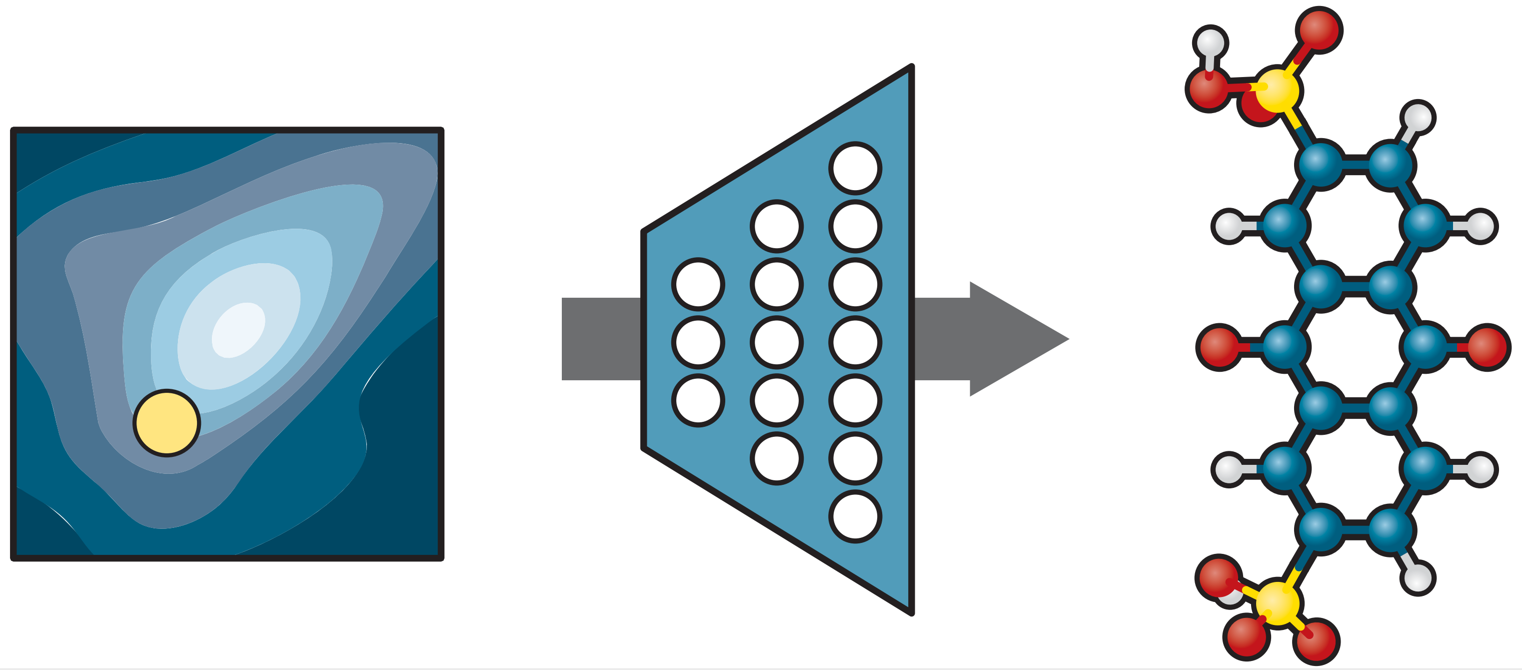
I found GAN to be less useful for quantitative scientific applications

Generative models and their physics genes



Generative AI for Science

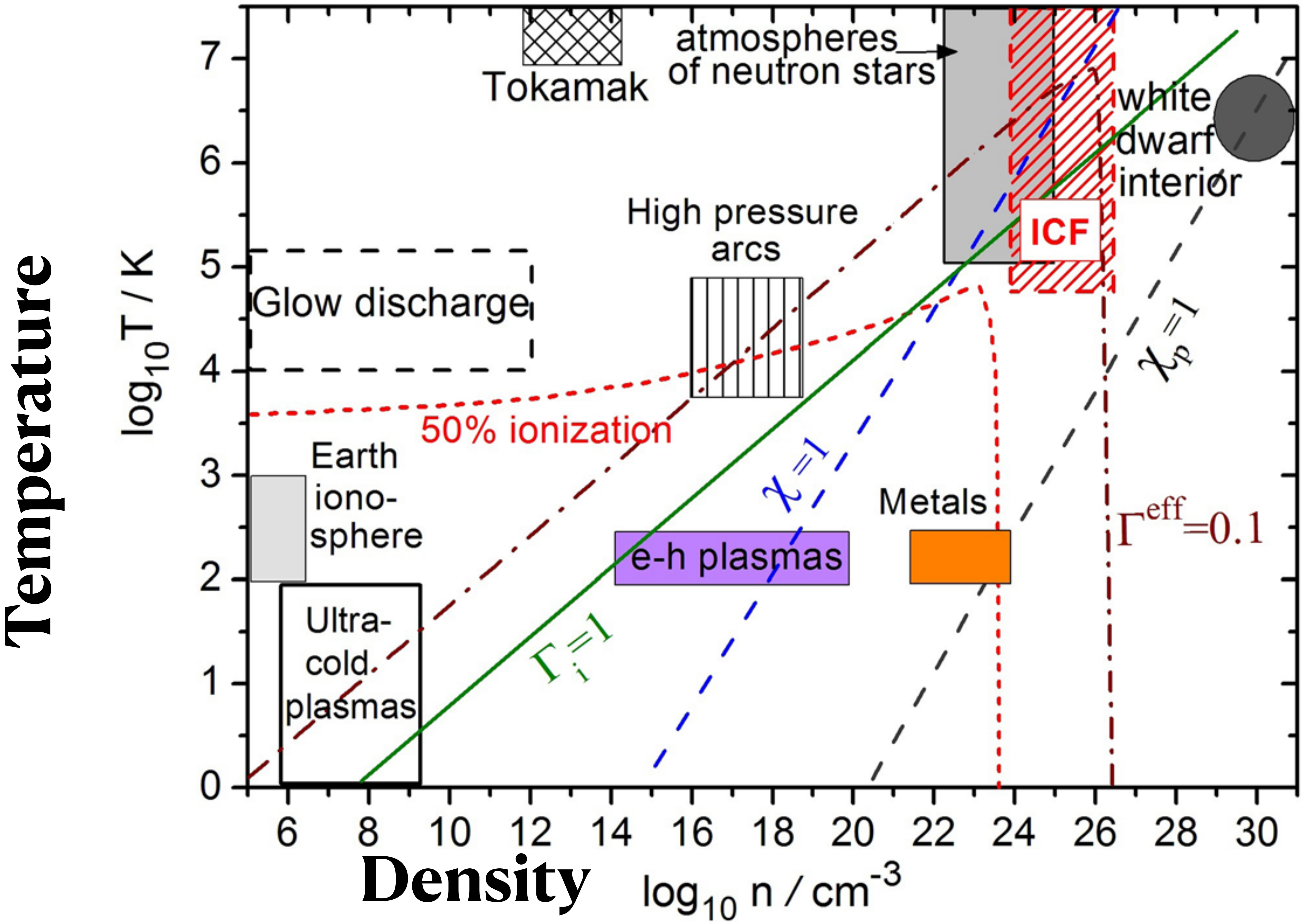
1. **How to Build a GPT-3 for Science** — Scientific language model

2.  Matter inverse design

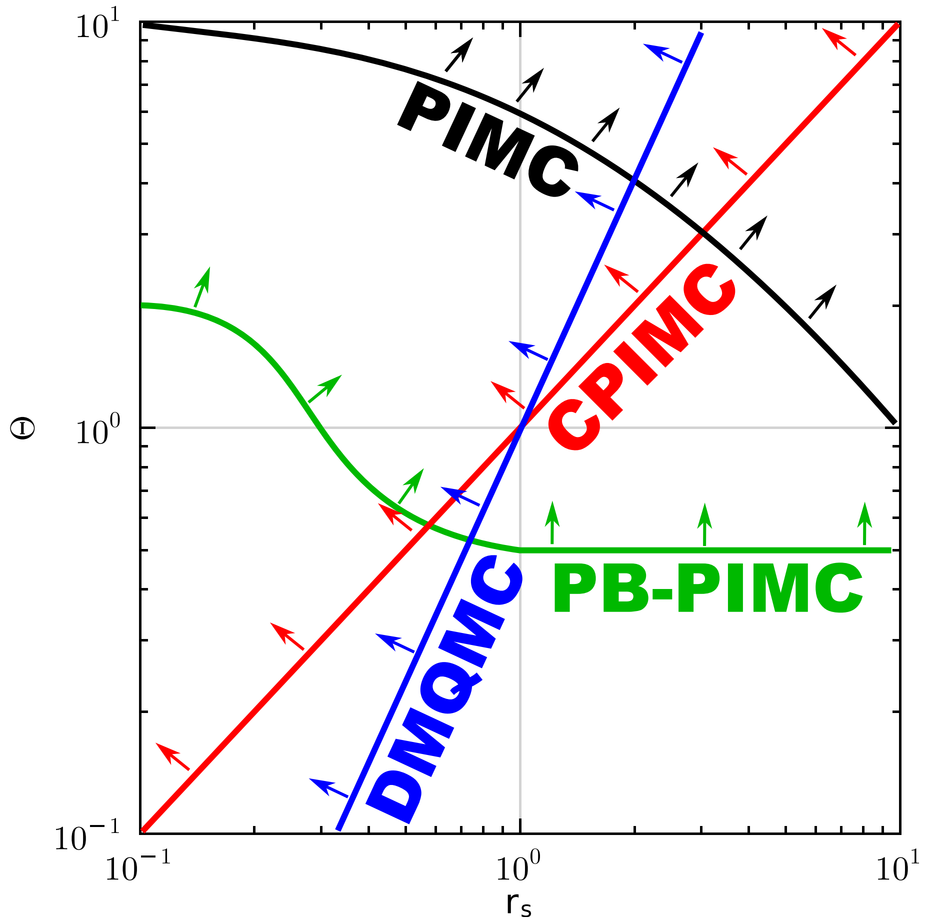
3. $$F = E - TS$$ Nature's cost function

Ab-initio study of quantum matters at T>0

$$H = -\sum_i \frac{\hbar^2}{2m_e}\nabla_i^2 - \sum_I \frac{\hbar^2}{2m_I}\nabla_I^2 - \sum_{I,i}\frac{Z_I e^2}{|R_I - r_i|} + \frac{1}{2}\sum_{i\neq j}\frac{e^2}{|r_i - r_j|} + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J e^2}{|R_I - R_J|}$$



Quantum Monte Carlo methods are limited by the "sign problem"



Dornheim et al, Phys. Plasmas '17
Bonitz et al, Phys. Plasmas '20

Classical world

Probability density p

Kullback-Leibler divergence

$$\mathbb{KL}\left(p\,||\,q\right)$$

Variational free-energy

$$F = \int dx \; \left[\frac{1}{\beta}p(\boldsymbol{x})\ln p(\boldsymbol{x}) + p(\boldsymbol{x})E(\boldsymbol{x})\right]$$

Quantum world

Density matrix ρ

Quantum relative entropy

$$S\left(\rho\,||\,\sigma\right)$$

Variational free-energy

$$F = \frac{1}{\beta}\mathrm{Tr}(\rho\ln\rho) + \mathrm{Tr}(\rho H)$$

Variational density matrix with *two* generative models

$$\min \quad F[\rho] = k_B T \,\mathrm{Tr}(\rho \ln \rho) + \mathrm{Tr}(H\rho)$$

Gibbs–Bogolyubov-Feynman-Delbrück–Molière

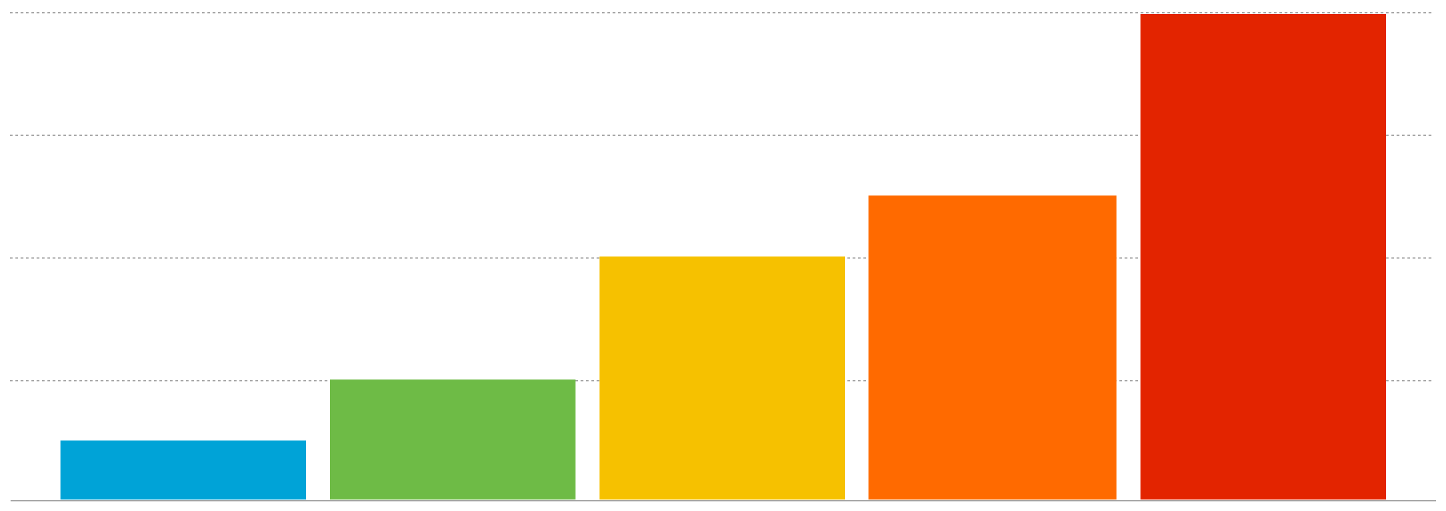
$$\mathrm{s.t.} \quad \mathrm{Tr}\rho = 1 \qquad \rho \succ 0 \qquad \rho^\dagger = \rho \qquad \langle \boldsymbol{X} | \rho | \boldsymbol{X}' \rangle = (-)^{\mathscr{P}} \langle \mathscr{P}\boldsymbol{X} | \rho | \boldsymbol{X}' \rangle$$

$$\boxed{\rho = \sum_n p_n |\Psi_n\rangle\langle\Psi_n|}$$

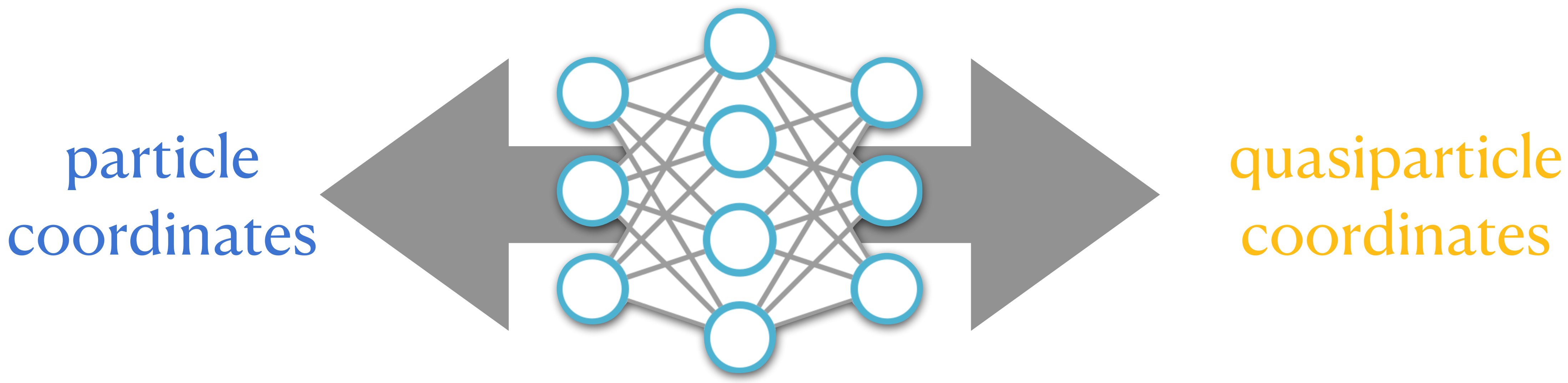
Xie, Zhang, LW, JML '22

Classical probability p_n

Quantum state basis $|\Psi_n\rangle$



Discrete probabilistic models
e.g. an autoregressive model



$\sqrt{\text{Normalizing flow}}$

c.f. Cranmer et al, 1904.05903, Saleh et al, 2308.16468

PHYSICAL REVIEW VOLUME 85, NUMBER 4 FEBRUARY 15, 1952

Point Transformations in Quantum Mechanics

BRYCE SELIGMAN DEWITT*
Ecole d'Eté de Physique Théorique de l'Université de Grenoble, Les Houches, Haute Savoie, France
(Received September 14, 1951)

An isomorphism is shown to exist between the group of point transformations in classical mechanics and a certain subgroup of the group of all unitary transformations in quantum mechanics. This isomorphism is

The unitary representations of the point-transformation group may be obtained by determining the infinitesimal generators of the group. An infinitesimal point transformation may be expressed in the form

$$x'^i = x^i + \epsilon\Lambda^i(x), \tag{3.7}$$

$$p_i' = p_i - \tfrac{1}{2}\epsilon[(\partial/\partial x^i)\Lambda^j(x), p_j]_+, \tag{3.8}$$

Coordinate transformation induces a unitary $e^{\frac{i}{2}[\Lambda^i(x), p_i]_+}$

ANNALS OF PHYSICS: **24**, 63–88 (1963)

Point Transformations and the Many Body Problem*

M. EGER† AND E. P. GROSS

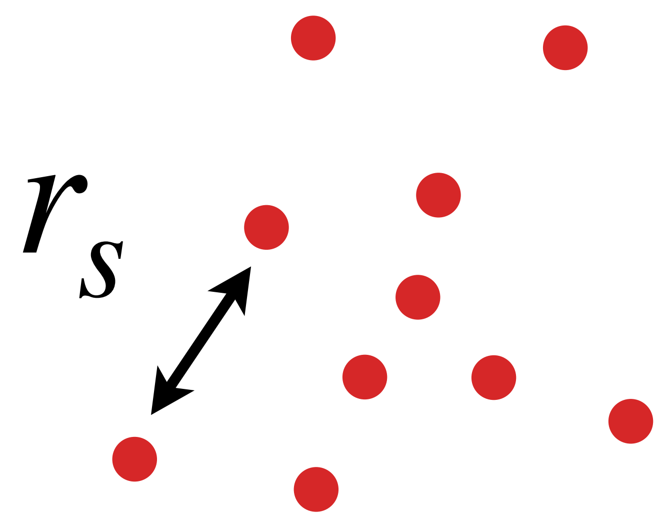
Brandeis University, Waltham, Massachusetts

An investigation is made of possible uses of many dimensional coordinate transformations in the quantum many-body problem. The transformed Hamiltonian is quadratic in the momenta with a space dependent metric. The original potential energy undergoes alteration and an additional "metric" potential energy appears. A relatively complete analysis of the transformed original potential is made, and the coordinate transformation can be used to suppress undesirable features of the original potential. For bosons one can attempt to directly map a complete set of noninteracting states onto approximate eigenstates of the system with interactions. Contact is made with a theory of weakly interacting bosons. In the general case it emerges that a given transformation uniquely fixes all the spatial correlation functions, which can be explicitly computed. The extended point transform can then be used as a link between diverse experimental quantities. The full use of the transformation to compute from first principles requires adequate approximations to the Jacobian and the inverse transform. These problems are not studied.

$\sqrt{\text{Normalizing flow}}$ materialize this dream

Example: uniform electron gas

$$H = -\sum_{i=1}^{N} \frac{\hbar^2 \nabla_i^2}{2m} + \sum_{i<j} \frac{e^2}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}$$



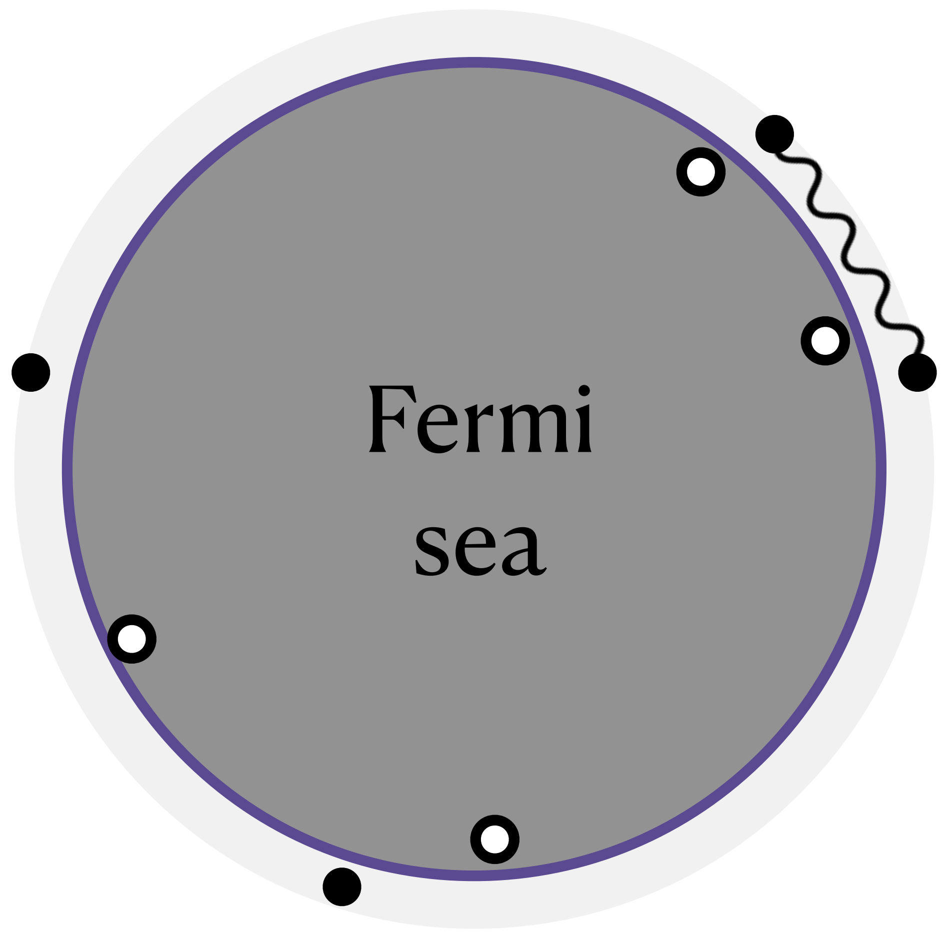
Fundamental model for metals ($2 < r_s < 6$)

Fermi liquid despite of non-perturbative r_s

$$\boxed{E_c[n] = \int d^3r\, n(\epsilon_c^{\mathrm{ueg}} + \cdots)}$$

Input to the density functional theory calculations

Deep generative models for the variational density matrix



Low-energy excited states are labeled in the same way as the ideal Fermi gas

$\boldsymbol{K} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_N\}$

$$\rho = \sum_{\boldsymbol{K}} p(\boldsymbol{K}) \left| \Psi_{\boldsymbol{K}} \rangle \langle \Psi_{\boldsymbol{K}} \right|$$

Normalized probability distribution

Orthonormal many-electron basis

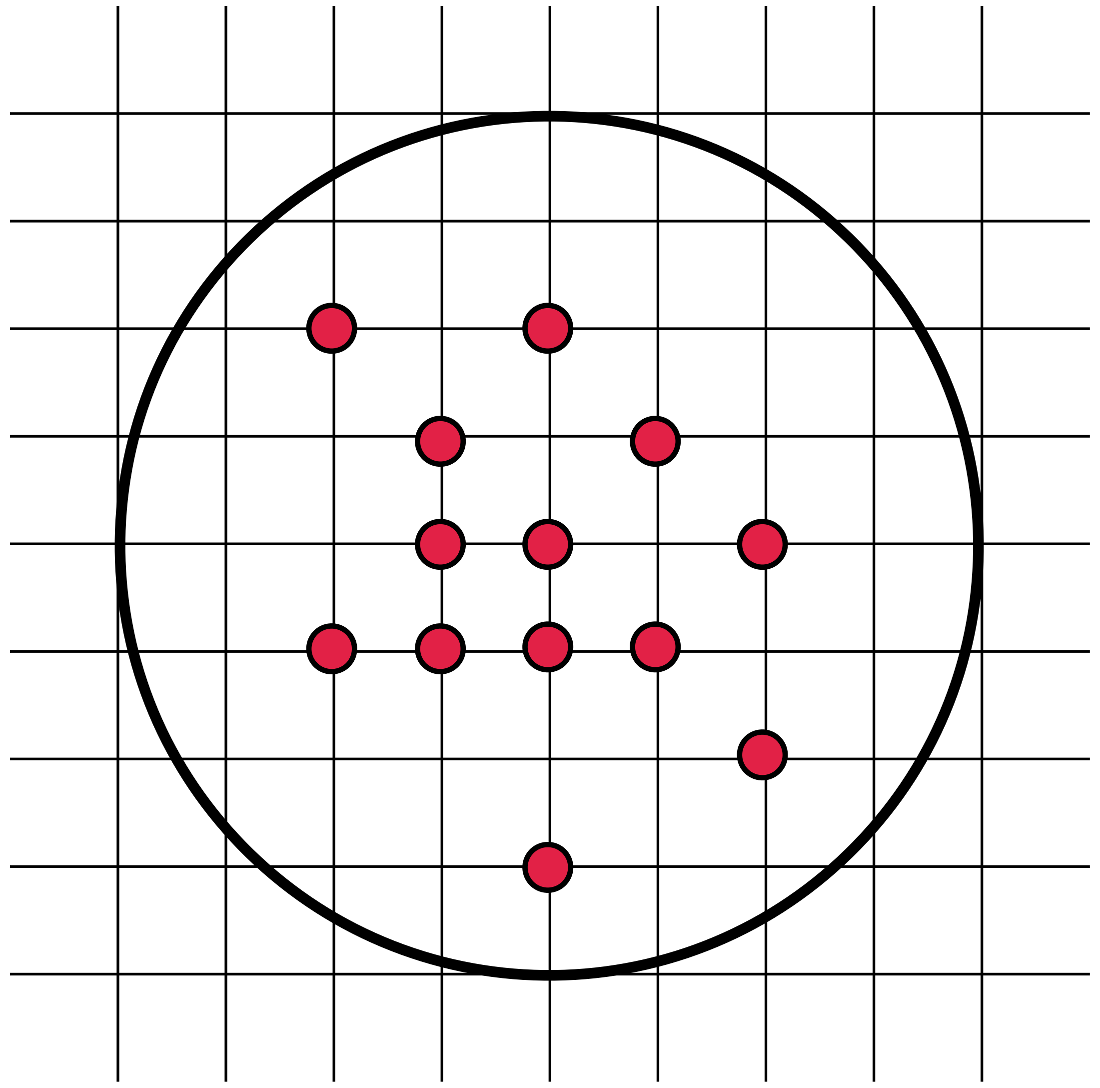
❶ $\sum_{\boldsymbol{K}} p(\boldsymbol{K}) = 1$

❷ $\langle \Psi_{\boldsymbol{K}} | \Psi_{\boldsymbol{K}'} \rangle = \delta_{\boldsymbol{K},\boldsymbol{K}'}$

Imposing physics constraints into deep generative models

❶ Autoregressive model for $p(\boldsymbol{K})$

Fermionic occupation in k-space



$$p(\boldsymbol{K}) = p(\boldsymbol{k}_1)p(\boldsymbol{k}_2\,|\,\boldsymbol{k}_1)p(\boldsymbol{k}_3\,|\,\boldsymbol{k}_1,\boldsymbol{k}_2)\cdots$$

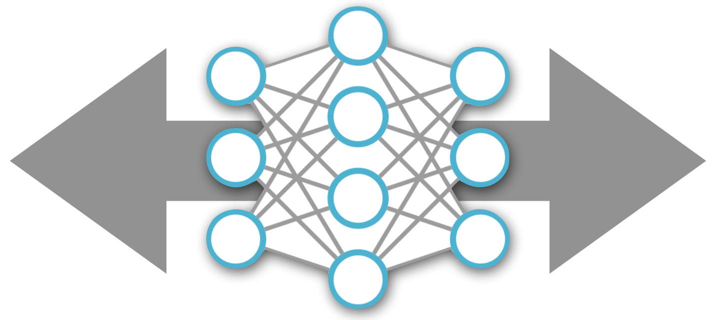
	Momentum distribution	Language
N	# of fermions	# of words
M	Momentum cutoff	Vocabulary
Space	$\binom{M}{N}$	M^N

Pauli exclusion: we are modeling a *set of words* with no repetitions and no order

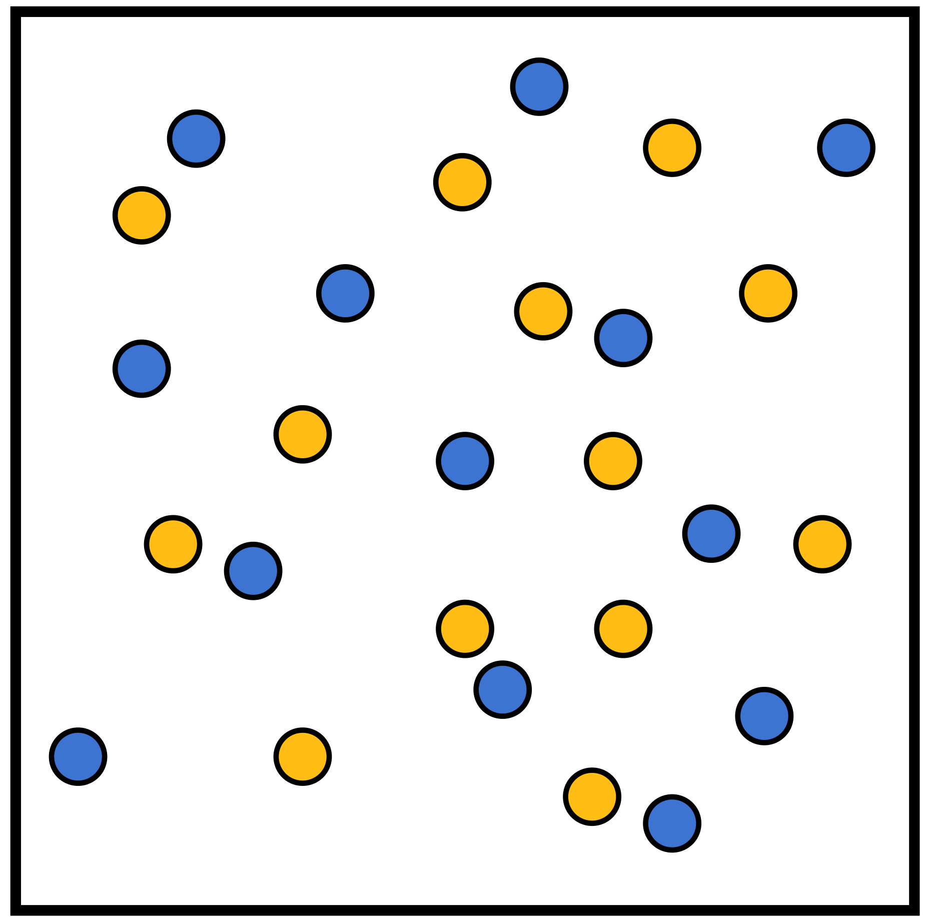
We use masked casual self-attention Vaswani et al 1706.03762; Alternative solution: Hibat-Allah et al, 2002.02793, Barrett et al, 2109.12606

$\sqrt{\text{Normalizing flow}}$ for $|\Psi_n\rangle$

Electron coordinates X



Z Quasi-particle coordinates



$$\Psi_K(X) = \frac{\det(e^{i k_i \cdot z_j})}{\sqrt{N!}} \cdot \left| \det \left(\frac{\partial Z}{\partial X} \right) \right|^{\frac{1}{2}}$$

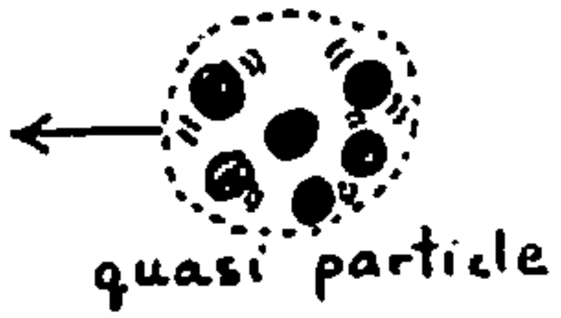
Orthonormal many-body states

Jacobian of the transformation

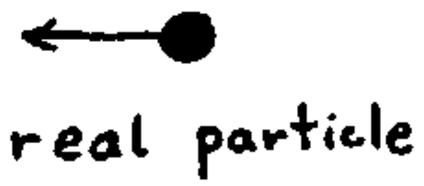
Fermion statistics: the flow should be permutation equivariant

we use FermiNet layer Pfau et al, 1909.02487, PRR ’20

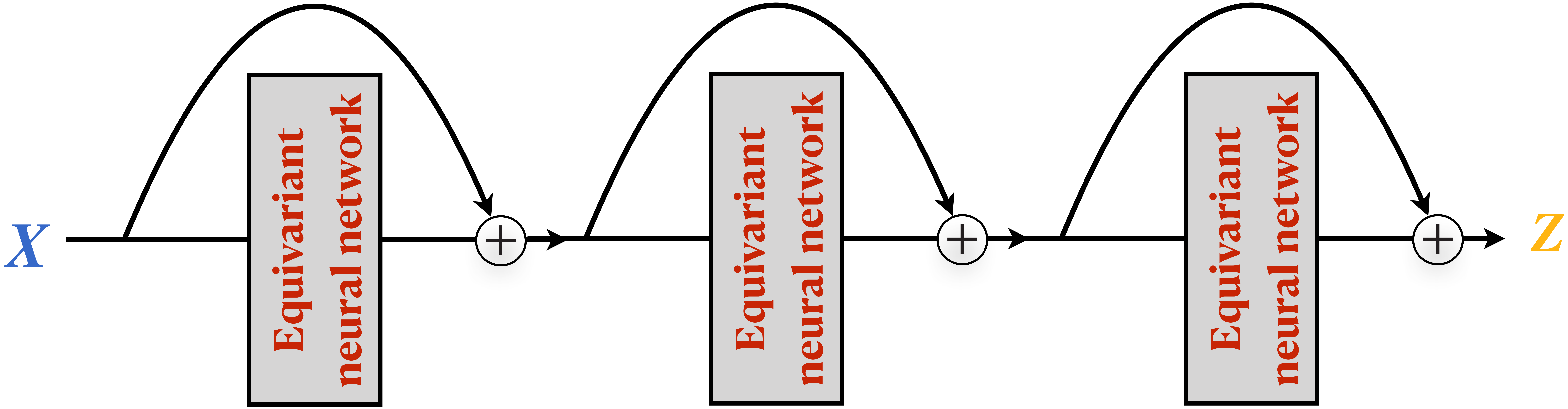
Feynman's backflow in the deep learning era



$$z_i = x_i + \sum_{j \neq i} \eta(|x_i - x_j|)(x_j - x_i)$$



Feynman & Cohen 1956
wavefunction for liquid Helium



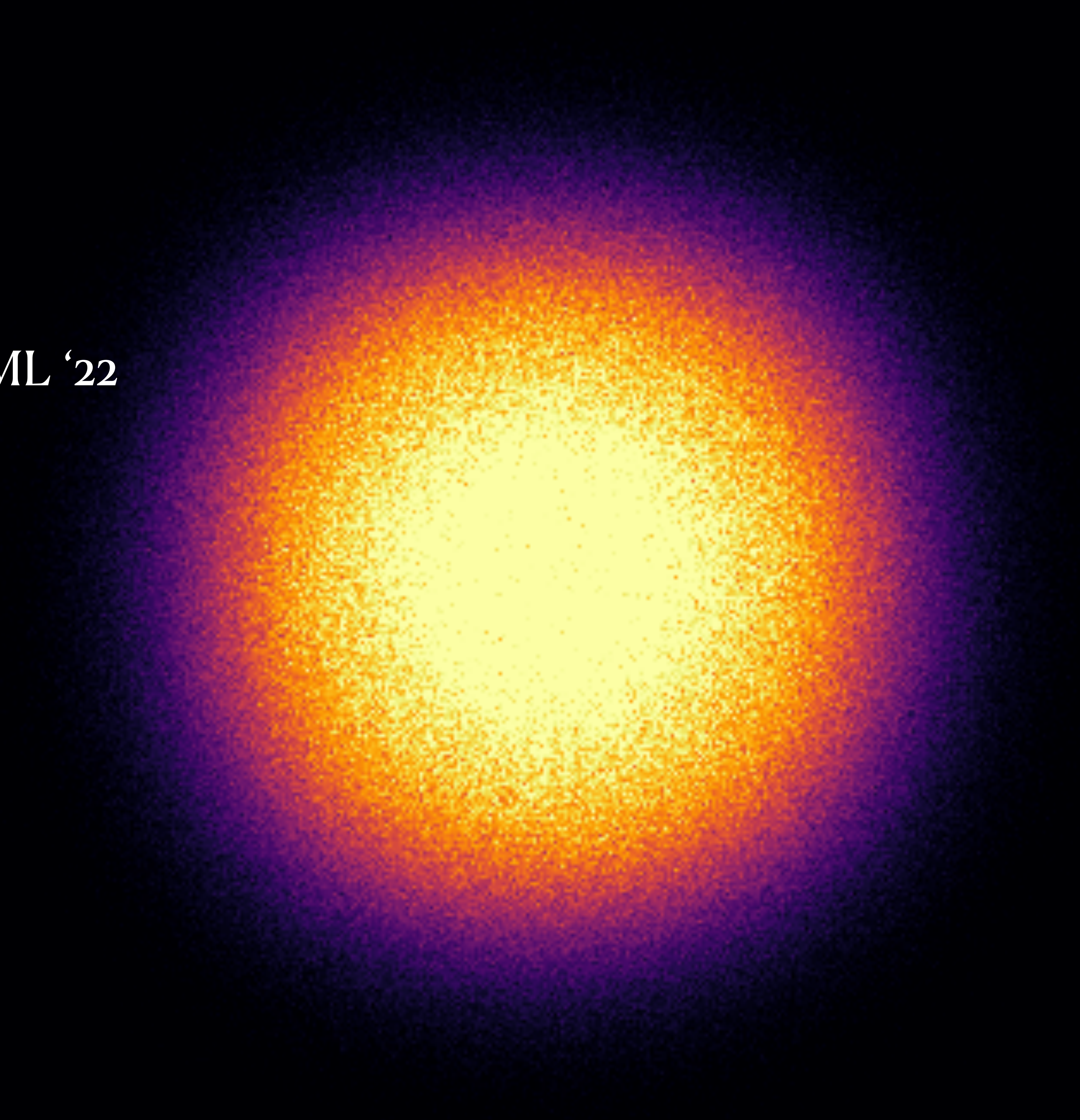
Iterative backflow → deep residual network → continuous normalizing flow

Taddei et al, PRB '15 E Commun. Math. Stat 17', Harbor el al 1705.03341, Lu et al 1710.10121, Chen et al, 1806.07366

Fermi Flow

Xie, Zhang, LW, 2105.08644, JML '22

github.com/fermiflow

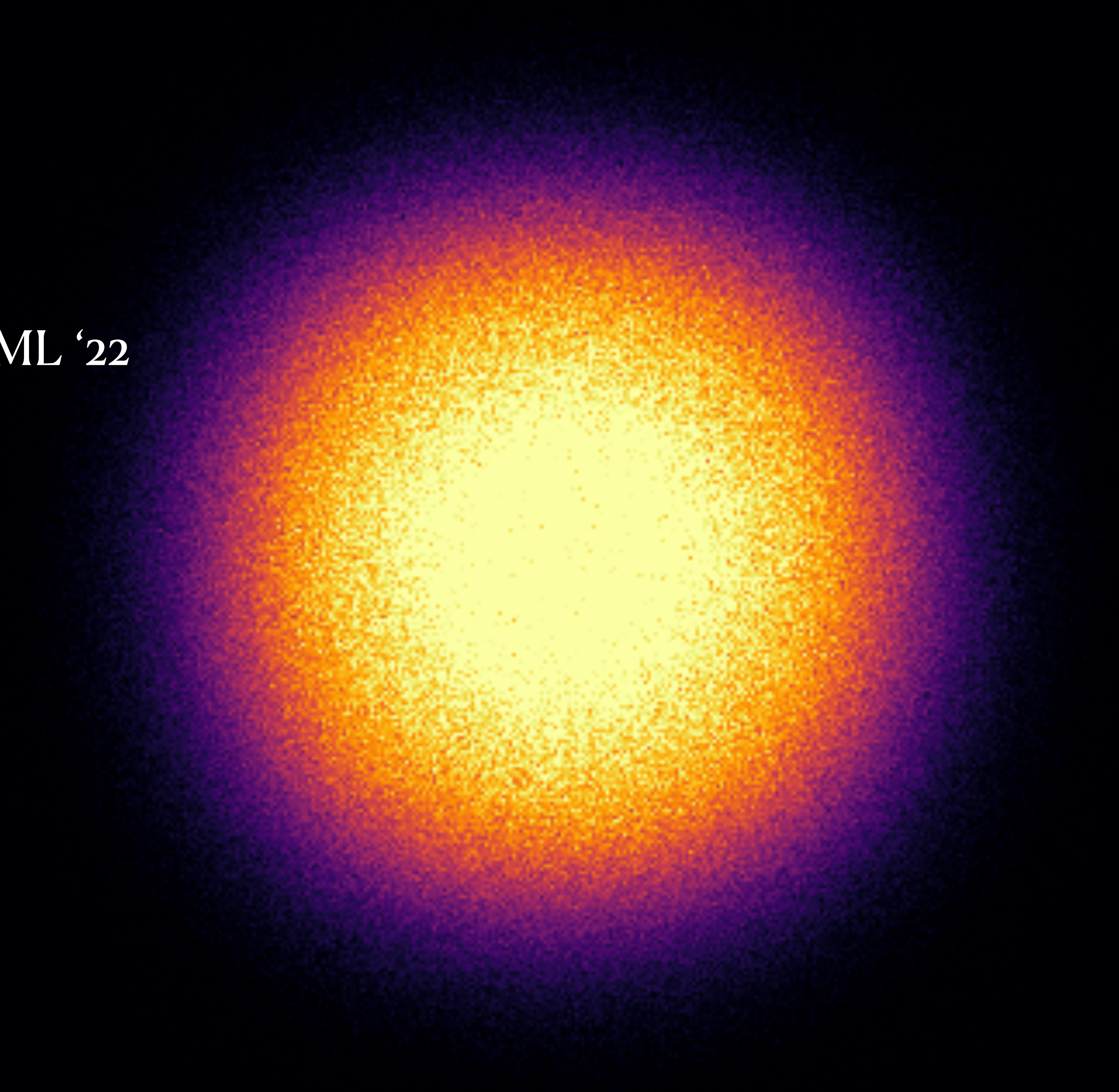


Continuous flow of electron density in a quantum dot

Fermi Flow

Xie, Zhang, LW, 2105.08644, JML '22

github.com/fermiflow



Continuous flow of electron density in a quantum dot

The objective function

$$F = \mathop{\mathbb{E}}_{\boldsymbol{K}\sim p(\boldsymbol{K})} \left[k_B T \ln p(\boldsymbol{K}) + \mathop{\mathbb{E}}_{\boldsymbol{X}\sim \left|\langle \boldsymbol{X} | \Psi_{\boldsymbol{K}}\rangle\right|^2} \left[\frac{\langle \boldsymbol{X} | H | \Psi_{\boldsymbol{K}}\rangle}{\langle \boldsymbol{X} | \Psi_{\boldsymbol{K}}\rangle} \right] \right]$$

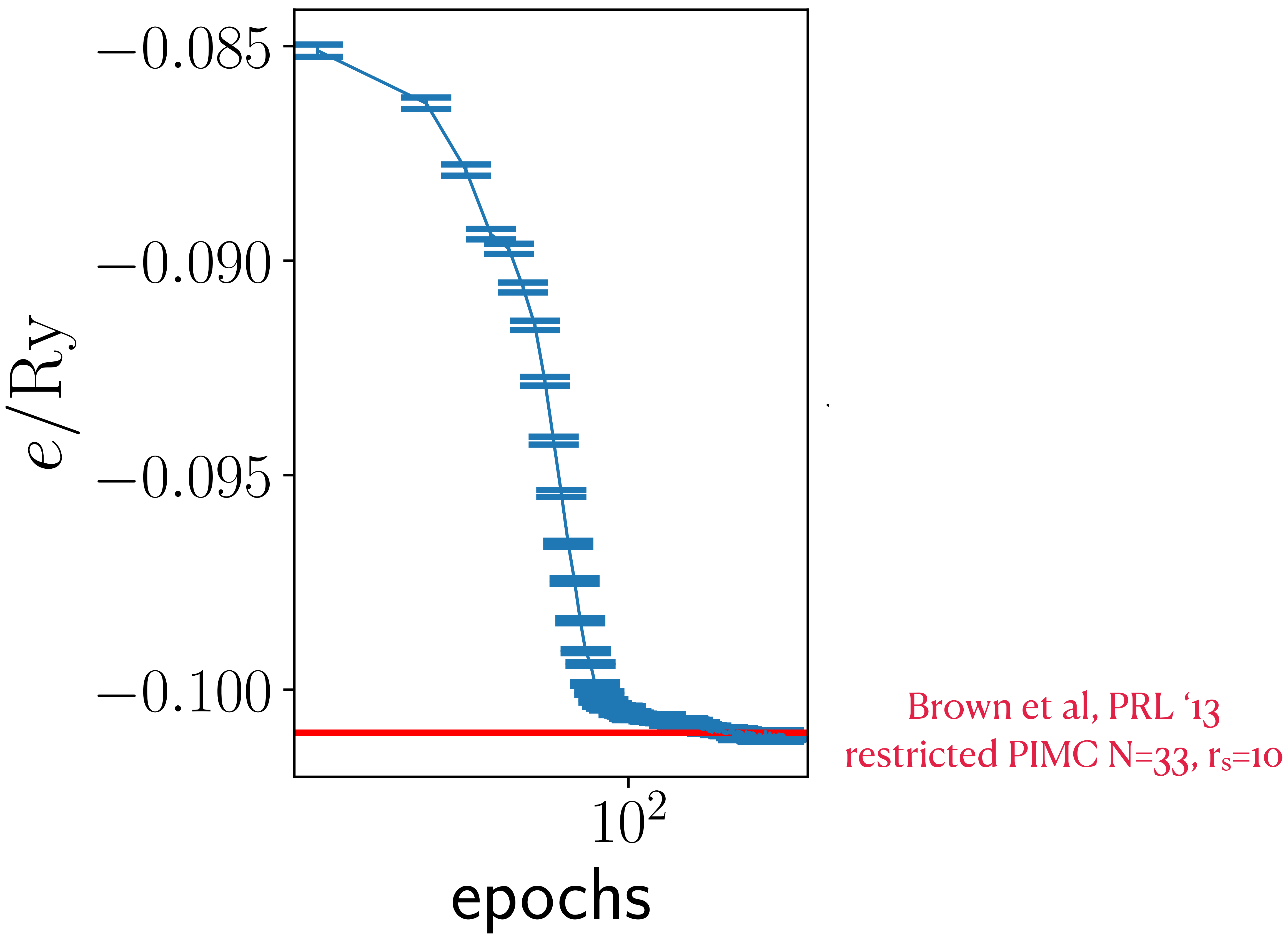
Boltzmann distribution

Born probability

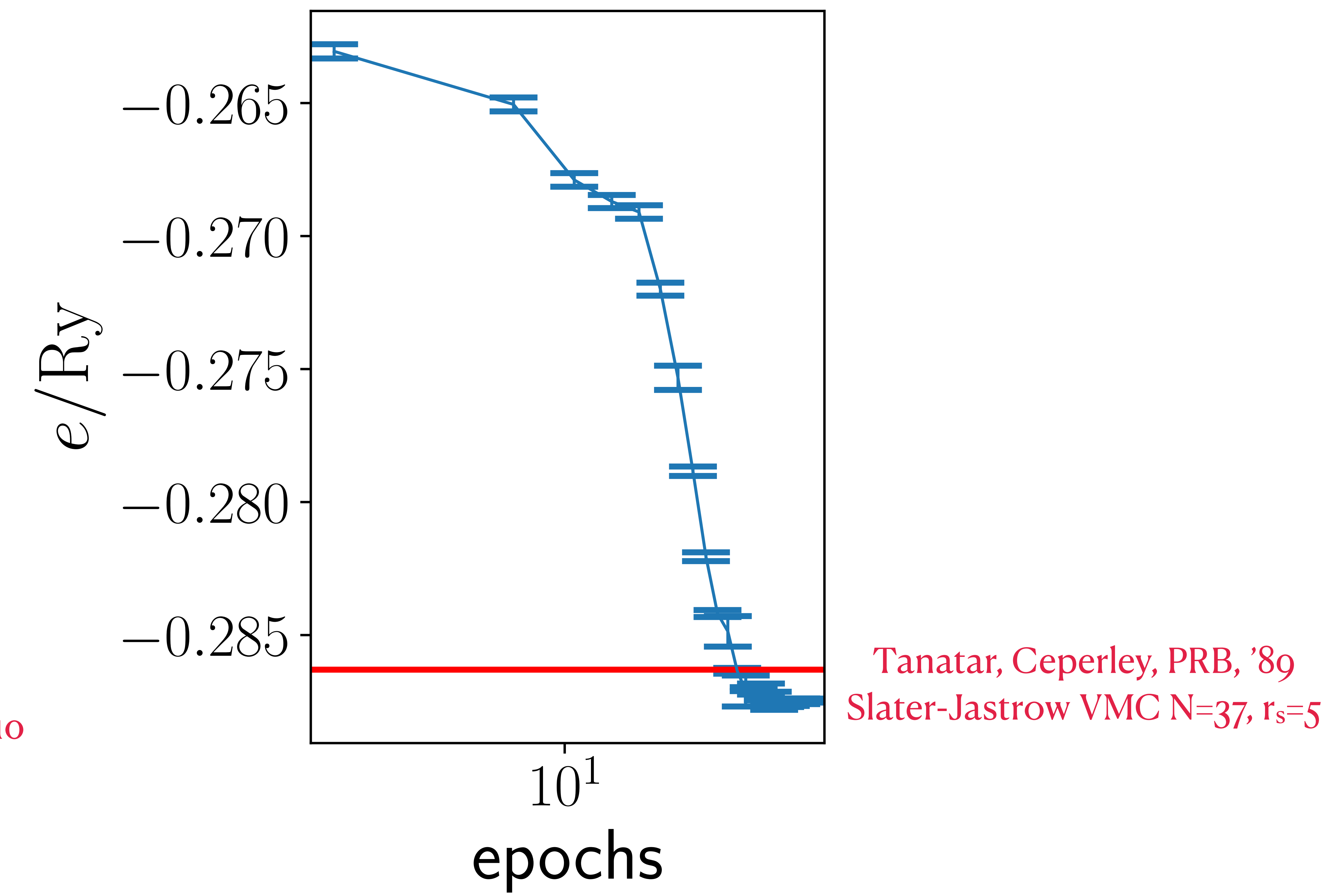
Jointly optimize $|\Psi_{\boldsymbol{K}}\rangle$ and $p(\boldsymbol{K})$ to minimize the variational free energy

Benchmarks on spin-polarized electron gases

3D electron gas T/T_F=0.0625



2D electron gas T=0



Application: m^* from low temperature entropy

Eich, Holzmann, Vignale, PRB '17

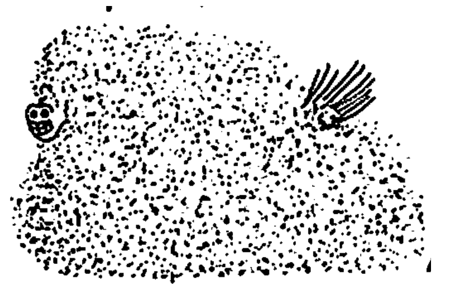
$$s = \frac{\pi^2 k_B}{3}\frac{m^*}{m}\frac{T}{T_F}$$

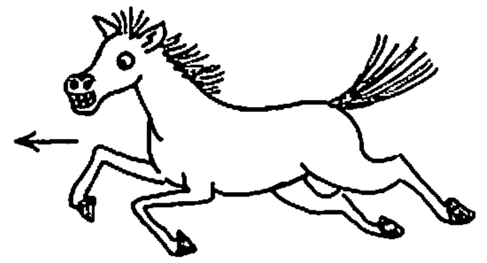
$$\Rightarrow \frac{m^*}{m} = \frac{s}{s_0}$$

interacting electrons

noninteracting electrons

Richard D. Mattuck
A Guide to Feynman Diagrams in the Many-body Problem



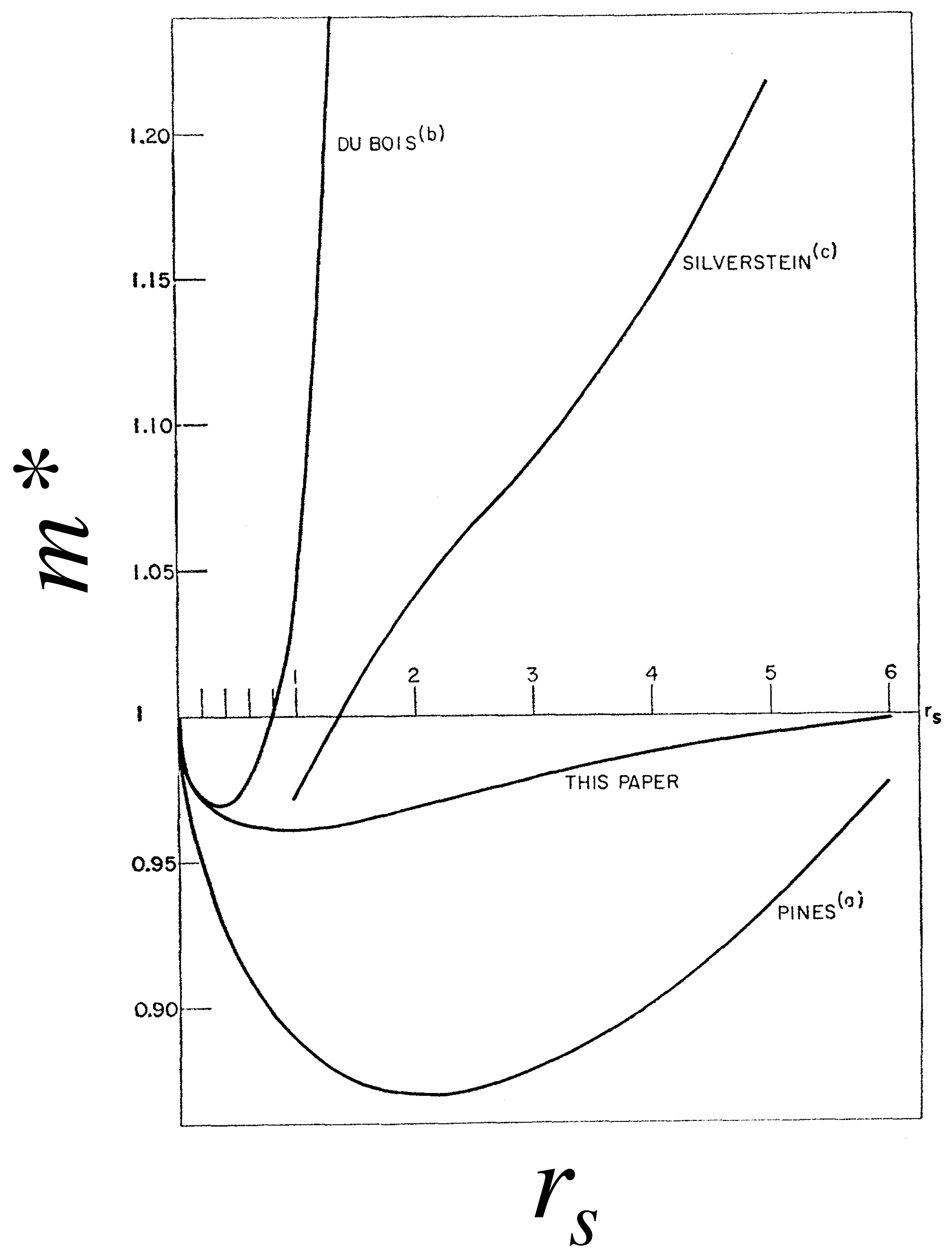


A fundamental quantity appears in nearly all physical properties of a Fermi liquid

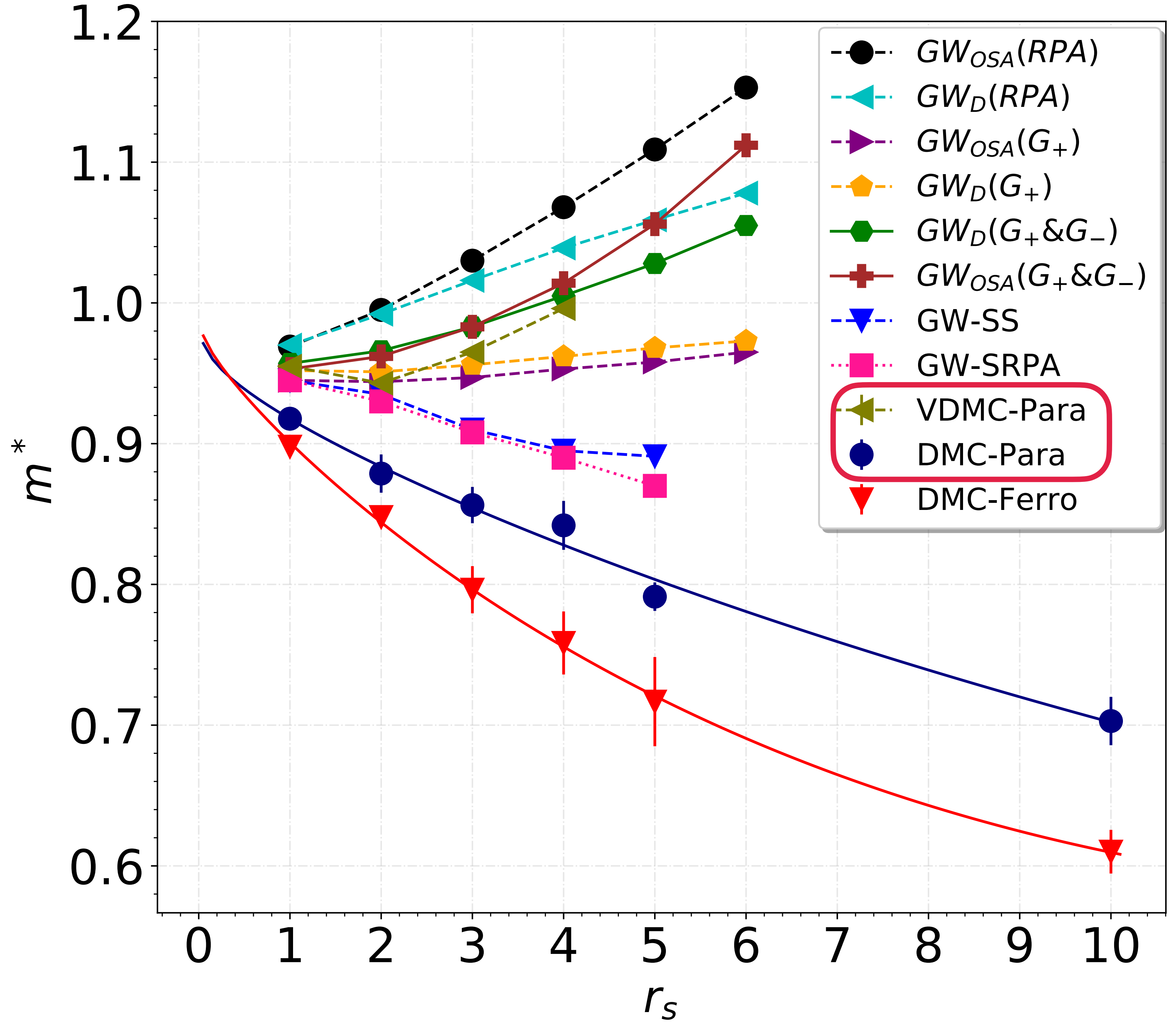
There have been debates despite its fundamental role and long history of study

Quasi-particles effective mass of 3d electron gas

Hedin Phy. Rev. 1965



Azadi, Drummond, Foulkes, PRL 2021



> 50 years of conflicting results !

Two-dimensional electron gas experiments

VOLUME 91, NUMBER 4 PHYSICAL REVIEW LETTERS week ending 25 JULY 2003

Spin-Independent Origin of the Strongly Enhanced Effective Mass in a Dilute 2D Electron System

A. A. Shashkin,* Maryam Rahimi, S. Anissimova, and S.V. Kravchenko
Physics Department, Northeastern University, Boston, Massachusetts 02115, USA

V.T. Dolgopolov
Institute of Solid State Physics, Chernogolovka, Moscow District 142432, Russia

T. M. Klapwijk
Department of Applied Physics, Delft University of Technology, 2628 CJ Delft, The Netherlands
(Received 13 January 2003; published 24 July 2003)

$m^*/m > 1$

PRL **101**, 026402 (2008) PHYSICAL REVIEW LETTERS week ending 11 JULY 2008

Effective Mass Suppression in Dilute, Spin-Polarized Two-Dimensional Electron Systems

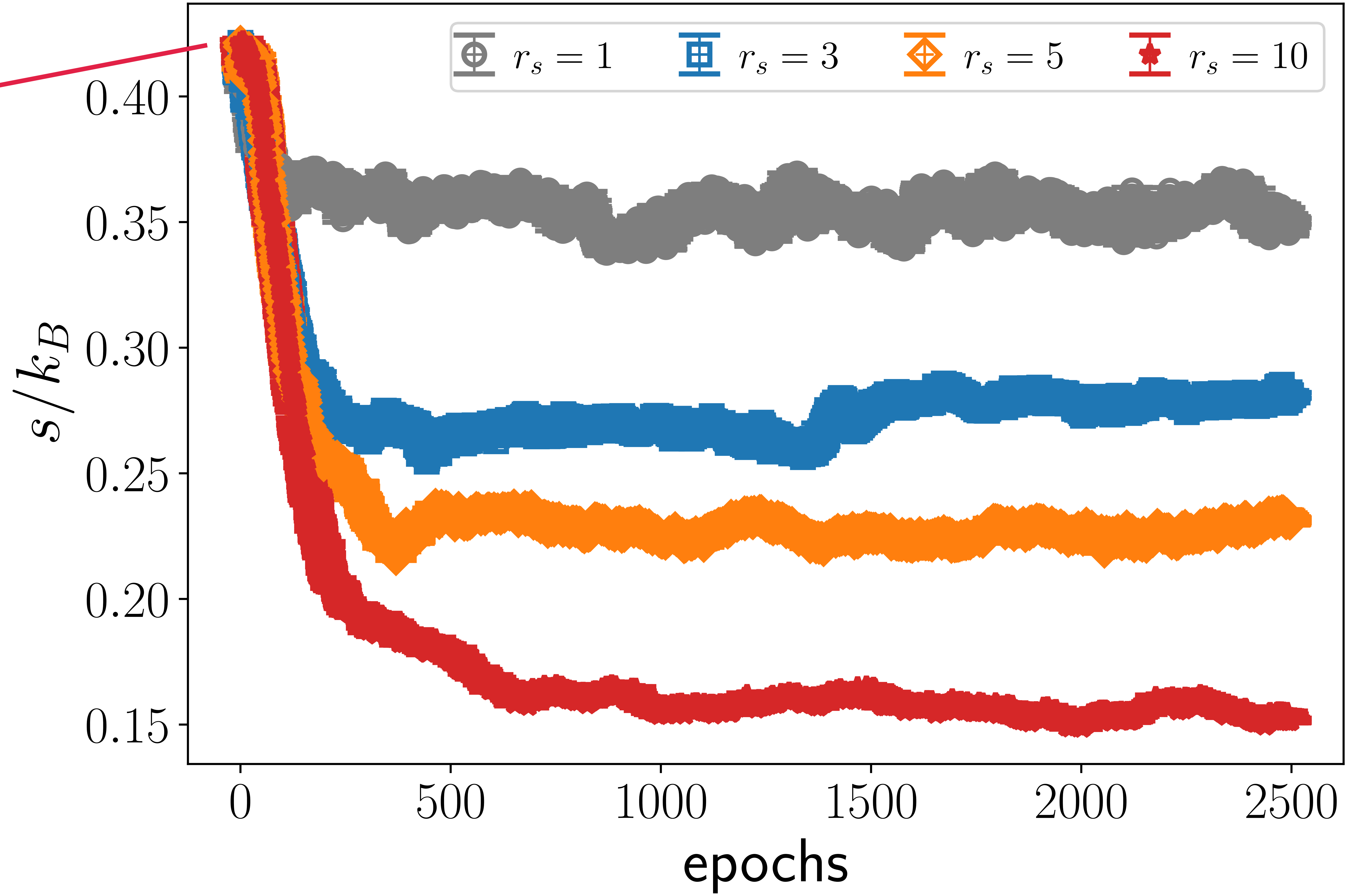
Medini Padmanabhan, T. Gokmen, N. C. Bishop, and M. Shayegan
Department of Electrical Engineering, Princeton University, Princeton, New Jersey 08544, USA
(Received 19 September 2007; published 7 July 2008)

$m^*/m < 1$

Layer thickness, valley, disorder, spin-orbit coupling...

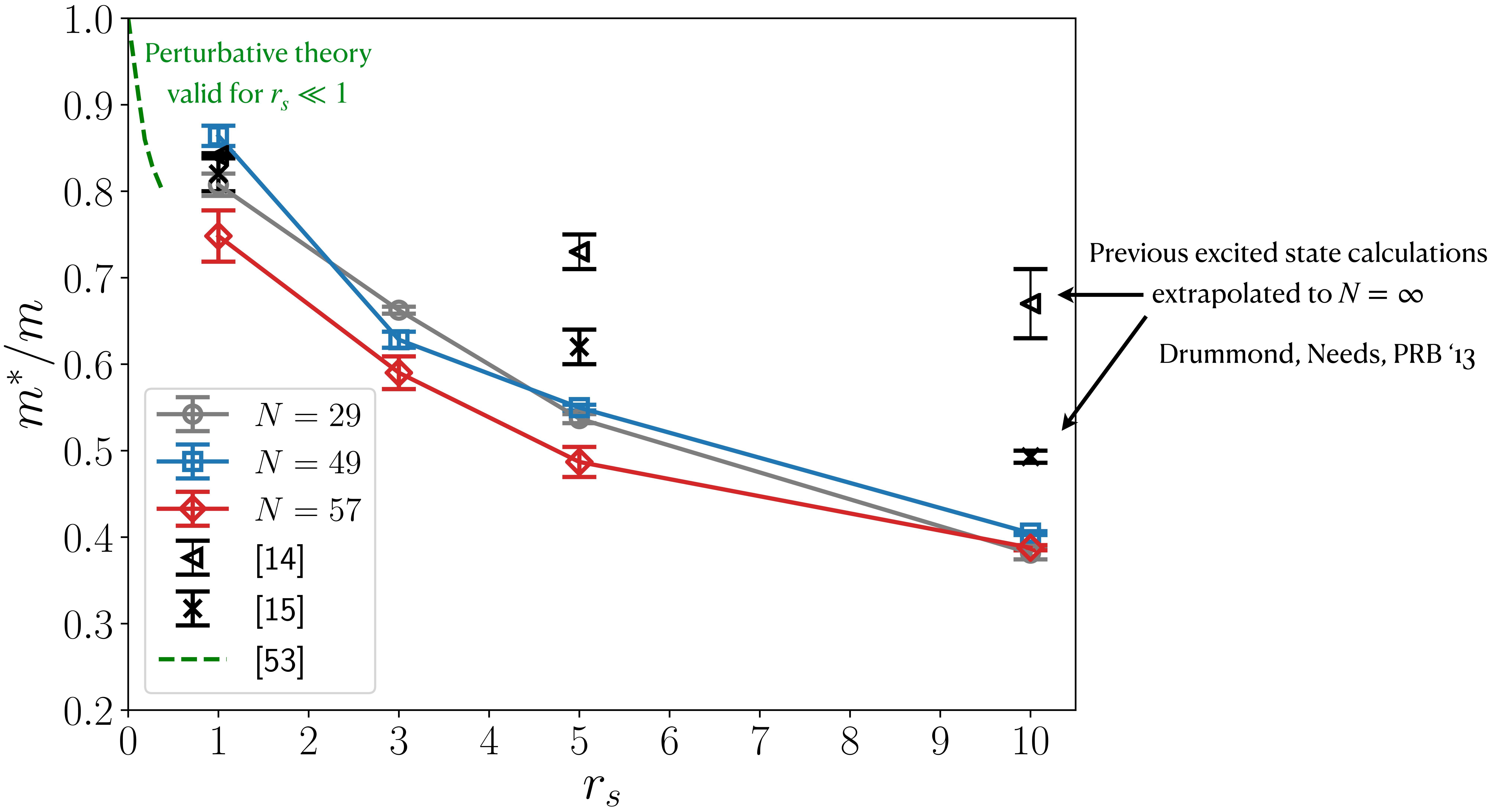
37 spin-polarized electrons in 2D @ T/T$_F$=0.15

Starting point:
Ideal gas entropy s_0



$$\frac{m^*}{m} = \frac{s}{s_0} < 1$$

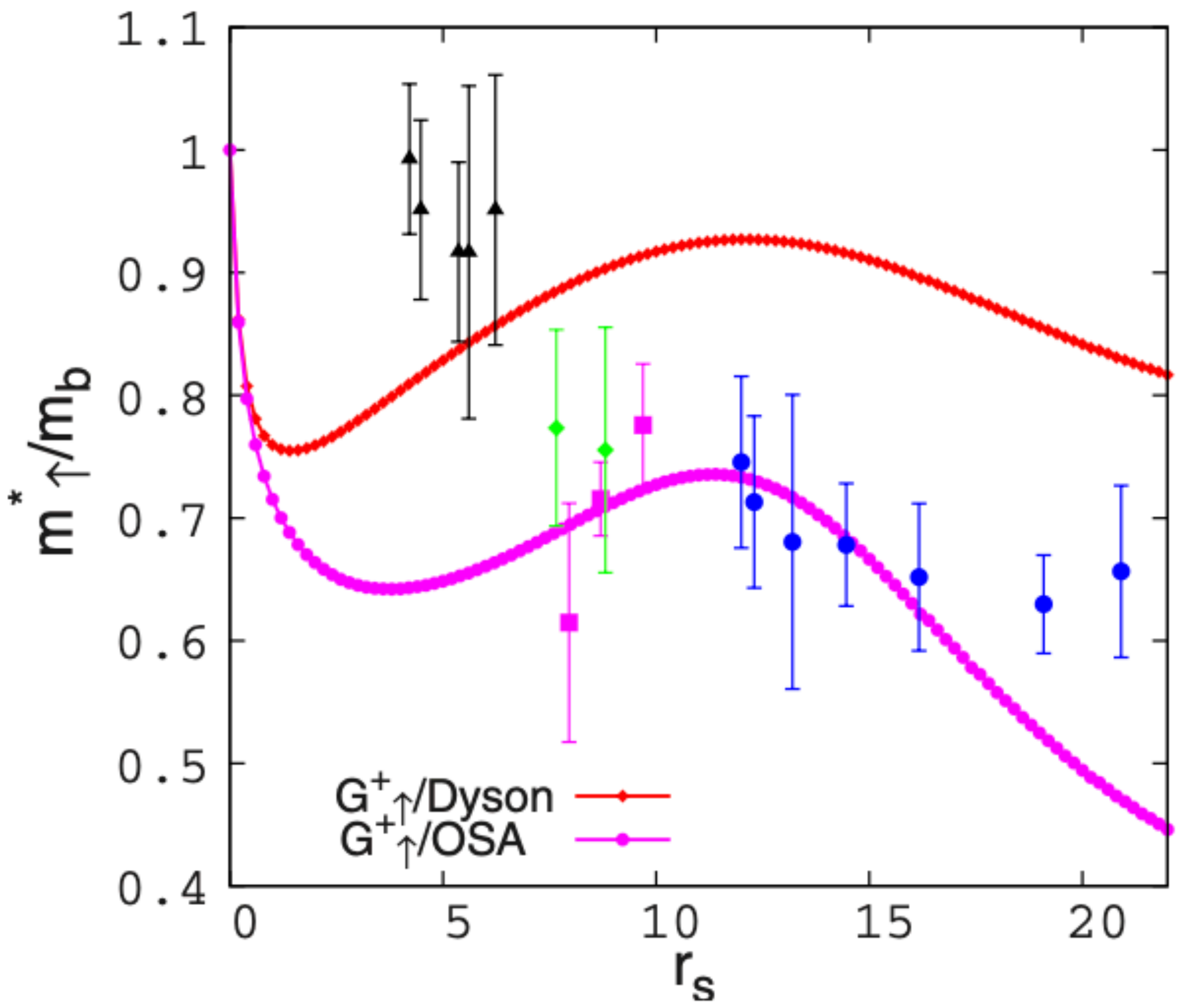
Effective mass of spin-polarized 2DEG



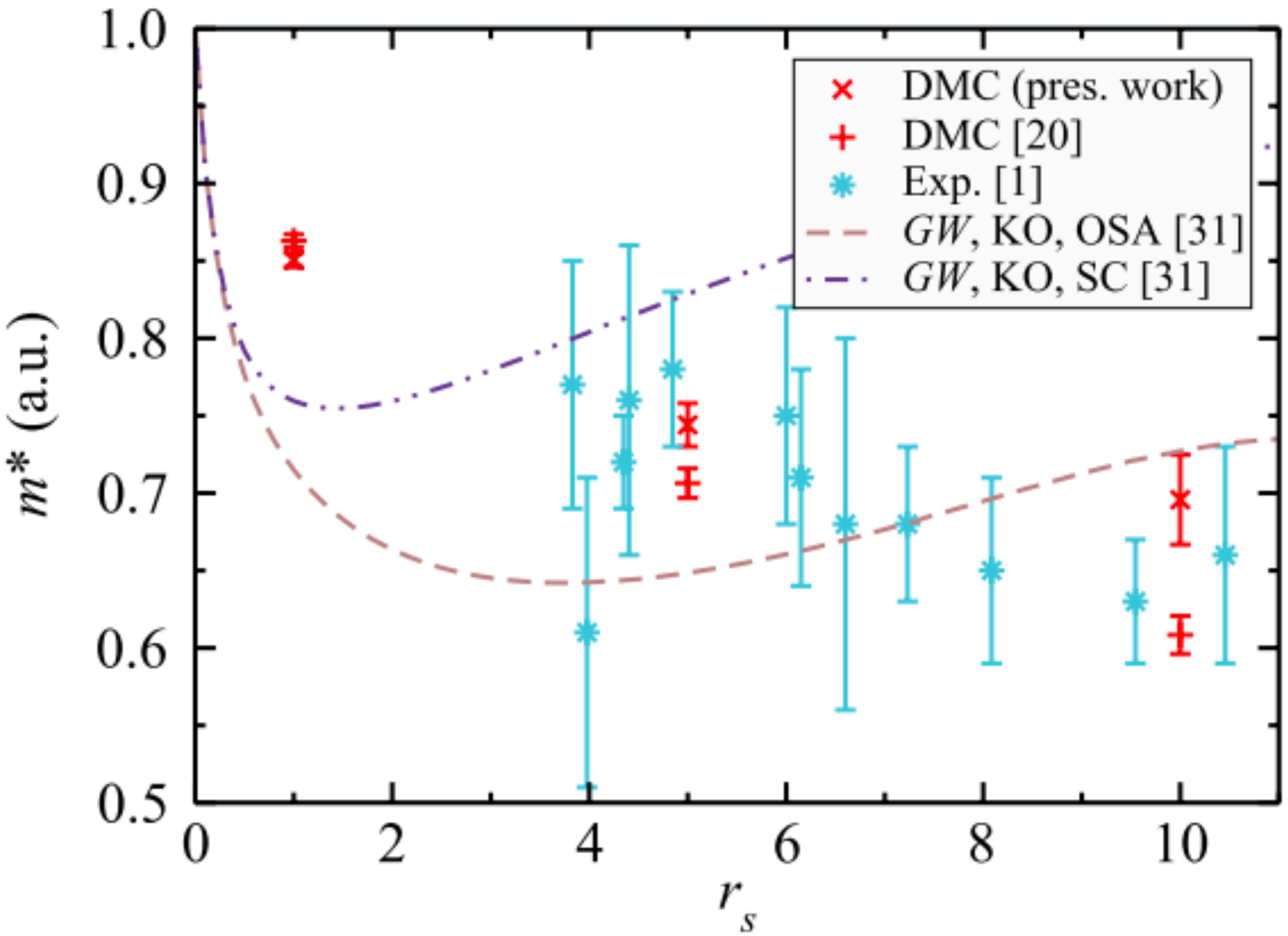
More pronounced suppression of m^* in the low-density strong-coupling region

Experiments on spin-polarized 2DEG

Asgari et al, PRB ‘09



Drommond, Needs, PRB’13



Quantum oscillation experiments
Padmanabhan et al, PRL ’08
Gokmen et al, PRB ‘09

Entropy measurement of 2DEG

ARTICLE

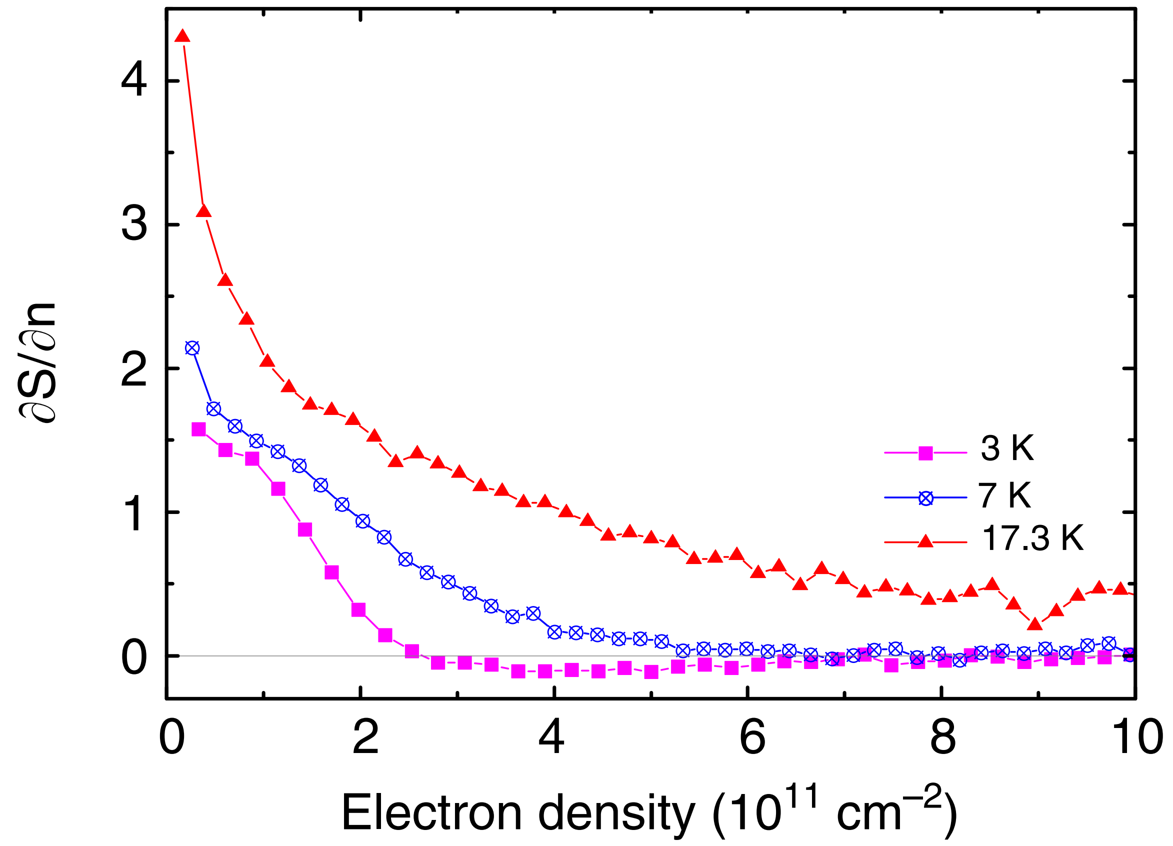
Received 16 May 2014 | Accepted 27 Apr 2015 | Published 23 Jun 2015

DOI: 10.1038/ncomms8298

Strongly correlated two-dimensional plasma explored from entropy measurements

A.Y. Kuntsevich[1,2], Y.V. Tupikov[3], V.M. Pudalov[1,2] & I.S. Burmistrov[2,4]

Maxwell relation $$\left(\frac{\partial S}{\partial n}\right)_T = -\left(\frac{\partial \mu}{\partial T}\right)_n$$



Next, directly compare computed entropy with the experiment

FAQs

Where to get training data ?

No training data. Data are self-generated from the generative model.

How do we know it is correct ?

Variational principle: lower free-energy is better.

Do I understand the "black box" model ?

a) I don't care (as long as it is sufficiently accurate).

b) $\ln p(\boldsymbol{K})$ contains the Landau energy functional

$$E[\delta n_k] = E_0 + \sum_k \epsilon_k \delta n_k + \frac{1}{2}\sum_{k,k'} f_{k,k'} \delta n_k \delta n_{k'}$$

$\boldsymbol{Z} \leftrightarrow \boldsymbol{X}$ illustrates adiabatic continuity.

“Using AI to accelerate scientific discovery” Demis Hassabis, co-founder and CEO of DeepMind 2021

Why now ?

Variational free-energy is a fundamental principle for T>0 quantum systems

However, it was under-exploited for solving practical problems (mostly due to intractable entropy for nontrivial density matrices)

Now, it is has became possible by integrating recent advances in generative models

The Universe as a generative model

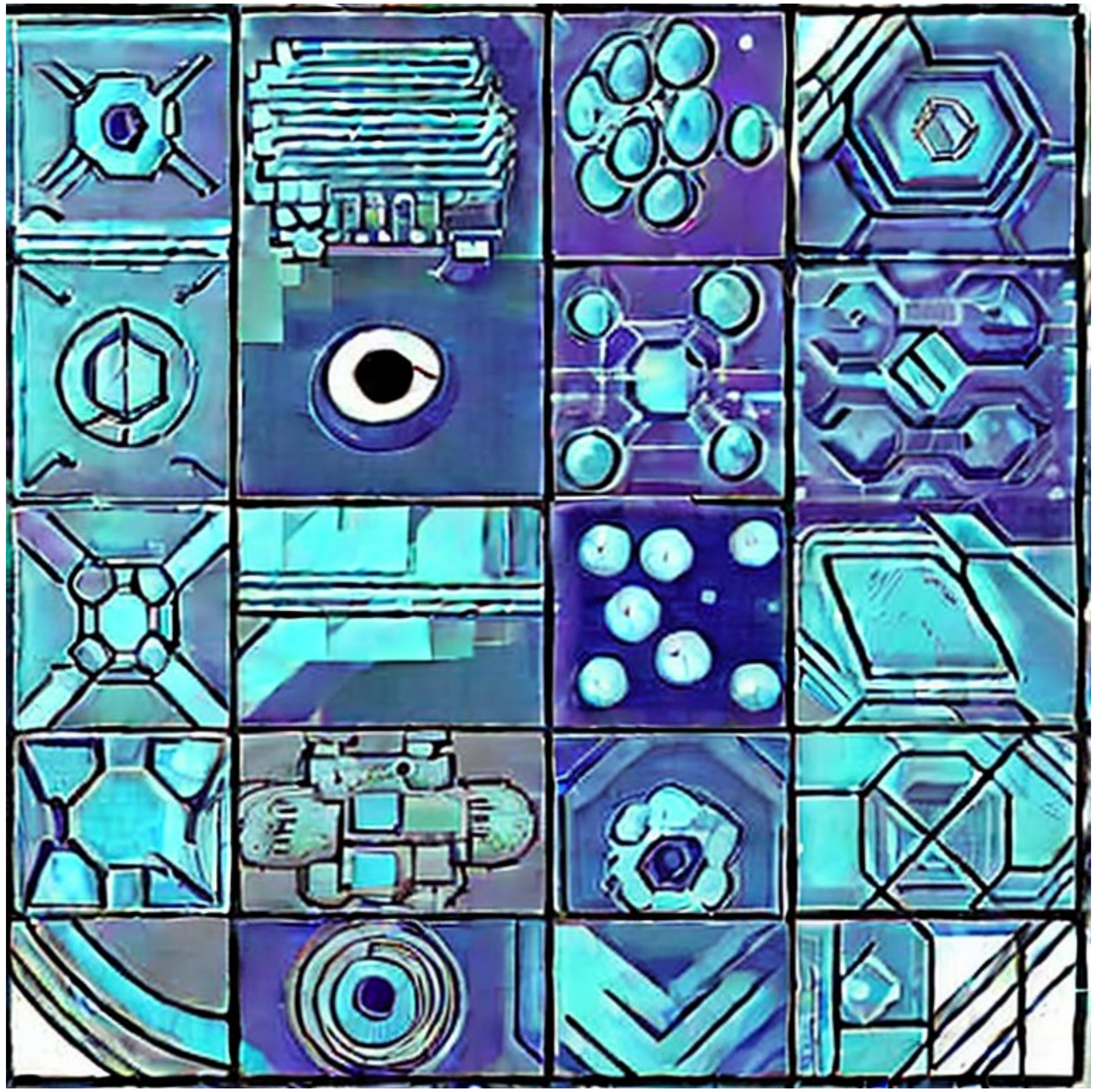
$$S = \int d^4x \sqrt{-g} \Big[\frac{m_p^2}{2} R - \frac{1}{4} F^a_{\mu\nu} F_a^{\mu\nu} + i \bar{\psi}^i \gamma^\mu D_\mu \psi^i + \left(\bar{\psi}^i_L V_{ij} \Phi \psi^j_R + h.c. \right) - |D_\mu \Phi|^2 - V(\Phi) \Big]$$

Thank you!

Discovering physical laws: learning the action

Solving physical problems: optimizing the action

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 ?



Machine learning for physicists

https://github.com/wangleiphy/ml4p

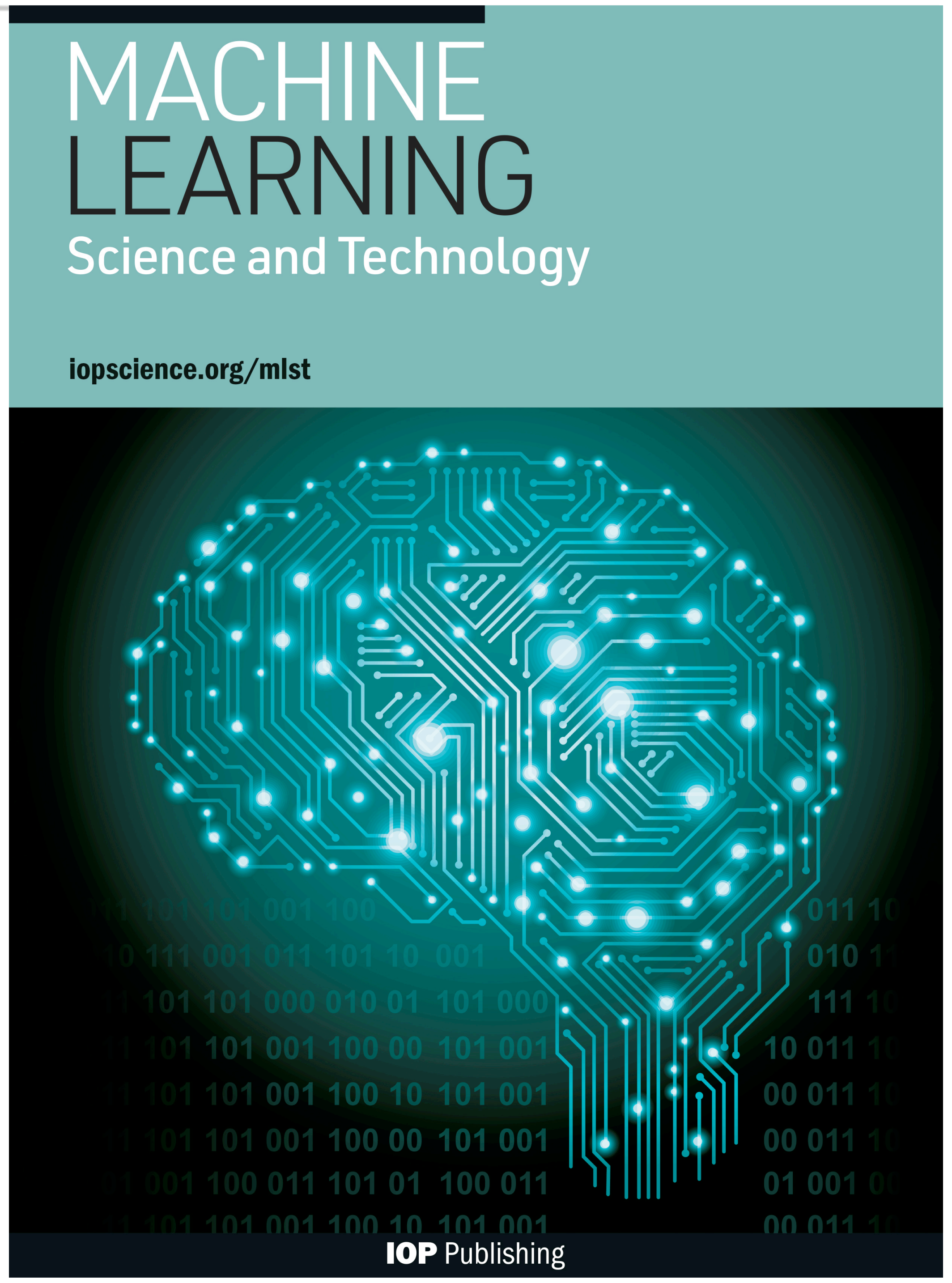
IOPscience | Journals | Books | Publishing Support | Login

Machine Learning: Science and Technology

Focus on Generative AI in Science

Guest Editors

Juan Felipe Carrasquilla, *Vector Institute, Canada*
Stephen R. Green, *University of Nottingham, UK*
Lei Wang, *Institute of Physics, CAS, China*
Linfeng Zhang, *DP Technology/AI for Science Institute, China*
Pan Zhang, *Institute of Theoretical Physics, CAS, China*



https://iopscience.iop.org/collections/mlst-230424-207