Unlocking the power of the variational free-energy principle with deep generative models

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

From the 2020 edition of this school

微分万物:深度学习的启示*

王 磊[1,2,†] 刘金国[3]
(1 中国科学院物理研究所 北京 100190)
(2 松山湖材料实验室 东莞 523808)
(3 哈佛大学物理系 剑桥 02138)

《物理》
2021年2月

generative model for sampling (and more)

quantum state tomography

experimental data analysis

high-throughput screening

force field

inverse design

density functional

differentiable programming

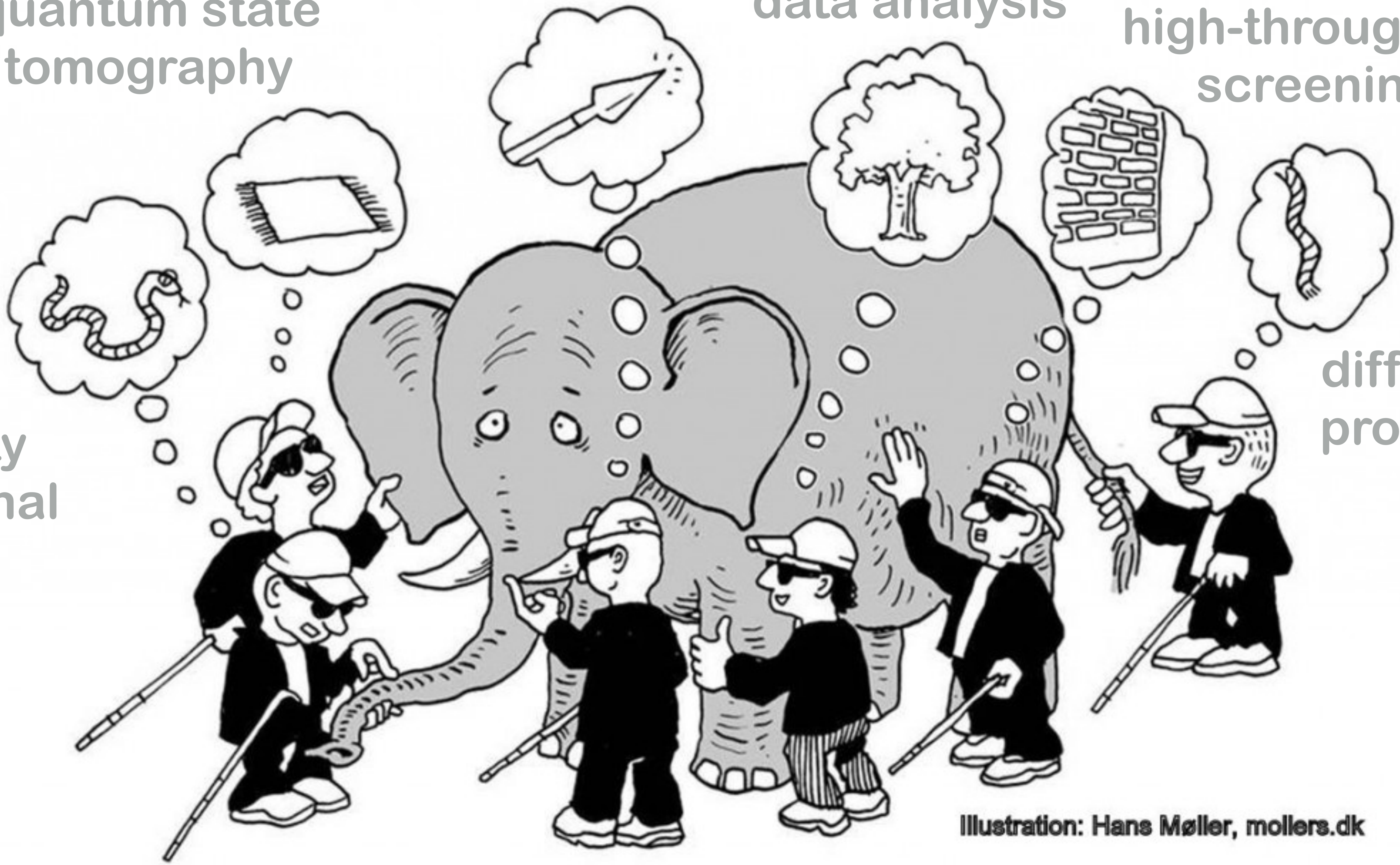
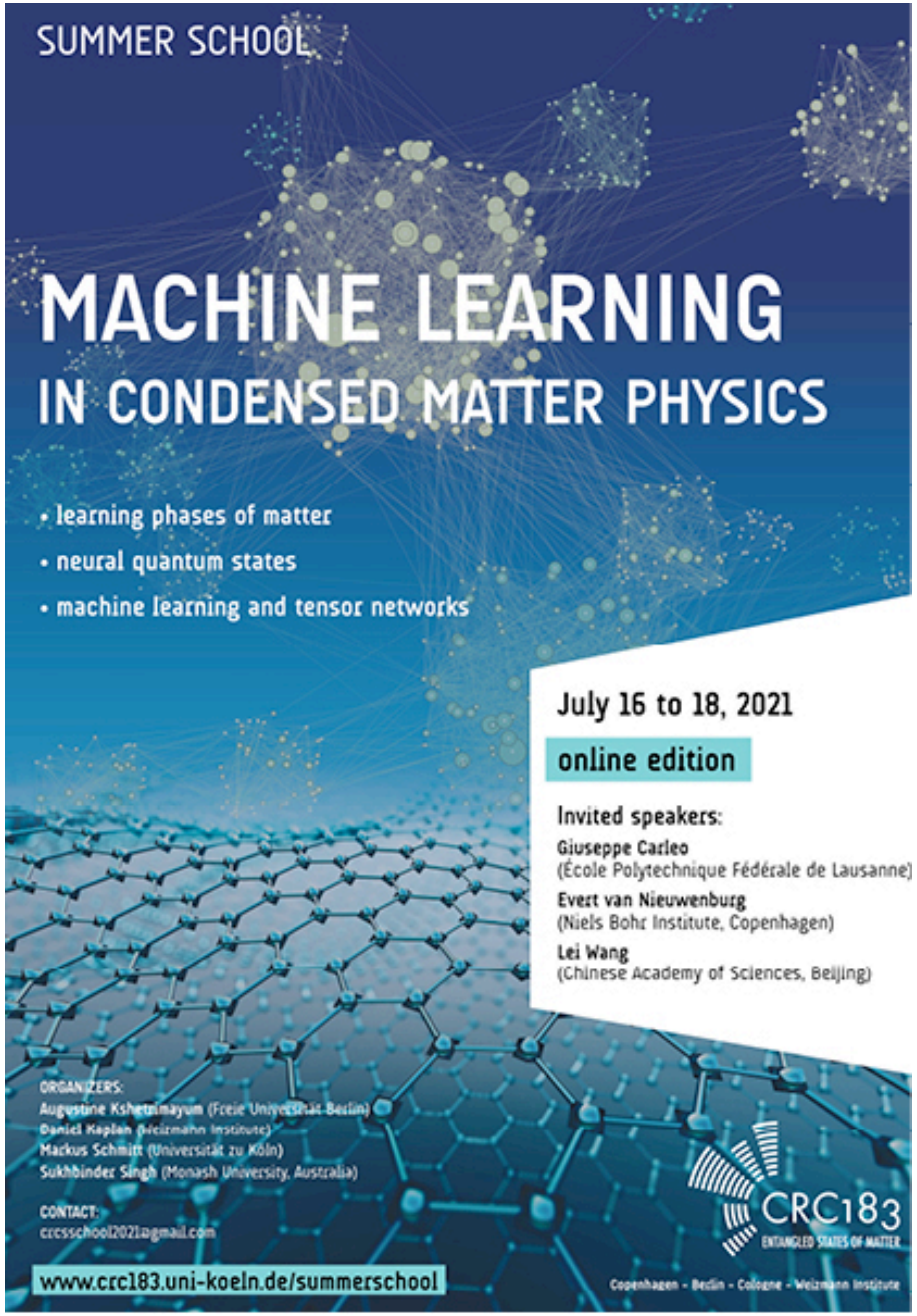


Illustration: Hans Møller, mollers.dk



Lei Wang 1.5h x 3

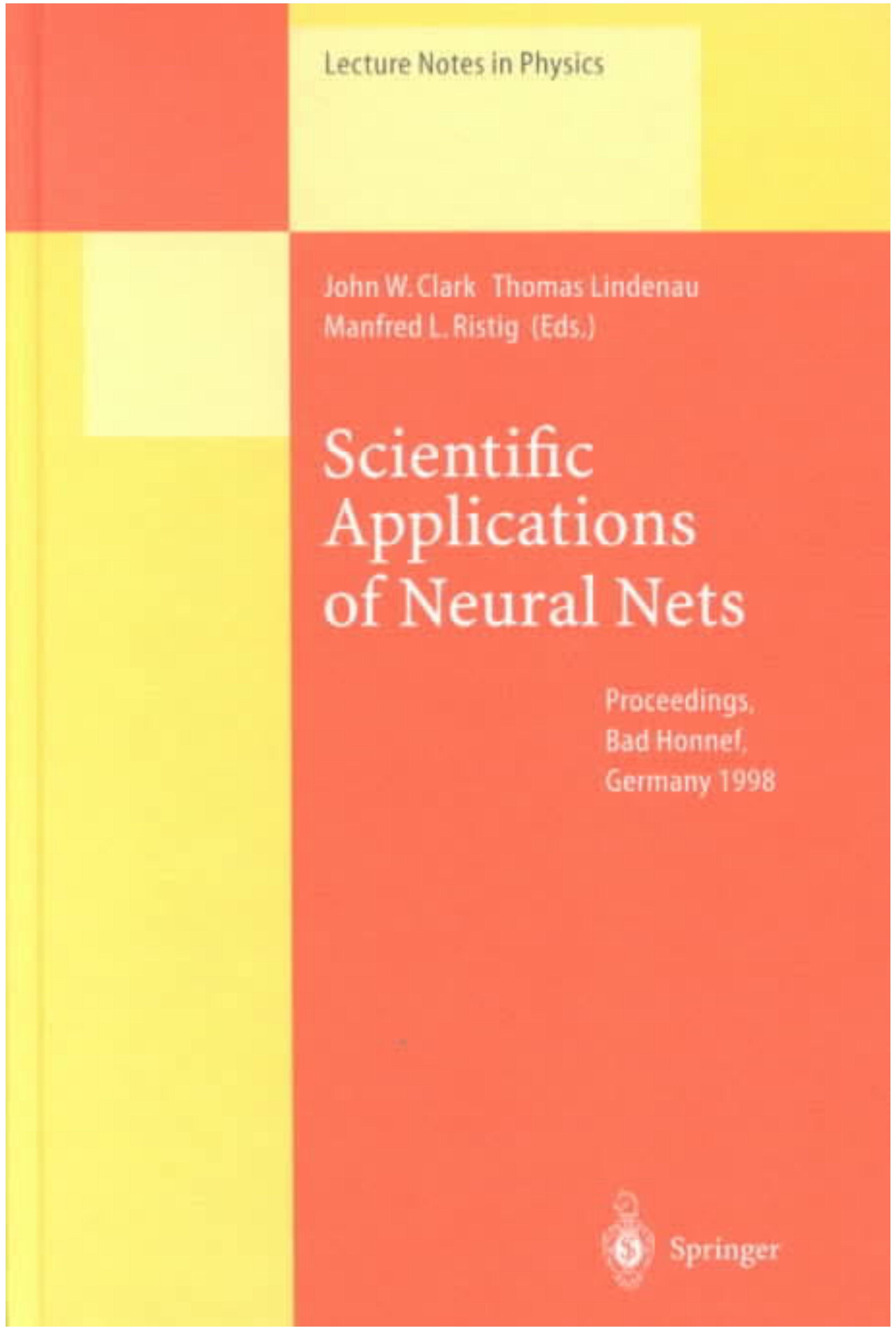
1. Scientific machine learning with and without data (overview talk)

2. Generative models (mostly normalizing flows)

3. Differentiable programming (for tensor networks and quantum circuits)

Recordings https://www.crc183.uni-koeln.de/summer-school-machine-learning/

AI for science, 24 years ago



8 Doing Science With Neural Nets: Pride and Prejudice

When neural networks re-emerged on the scene in the mid-80s as a new and glamorous computational paradigm, the initial reaction in some sectors of the scientific community was perhaps too enthusiastic and not sufficiently critical. There was a tendency on the part of practitioners to oversell the powers of neural-network or "connectionist" solutions relative to conventional techniques – where conventional techniques can include both traditional theory-rich modeling and established statistical methods. The last five years have seen a correction phase, as some of the practical limitations of neural-network approaches have become apparent, and as scientists have become better acquainted with the wide array of advanced statistical tools that are currently available.

Why now, again ?

What has changed ?

What has not ?

A hint from the Deep Learning Book

"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."

Deep learning is more than fitting!

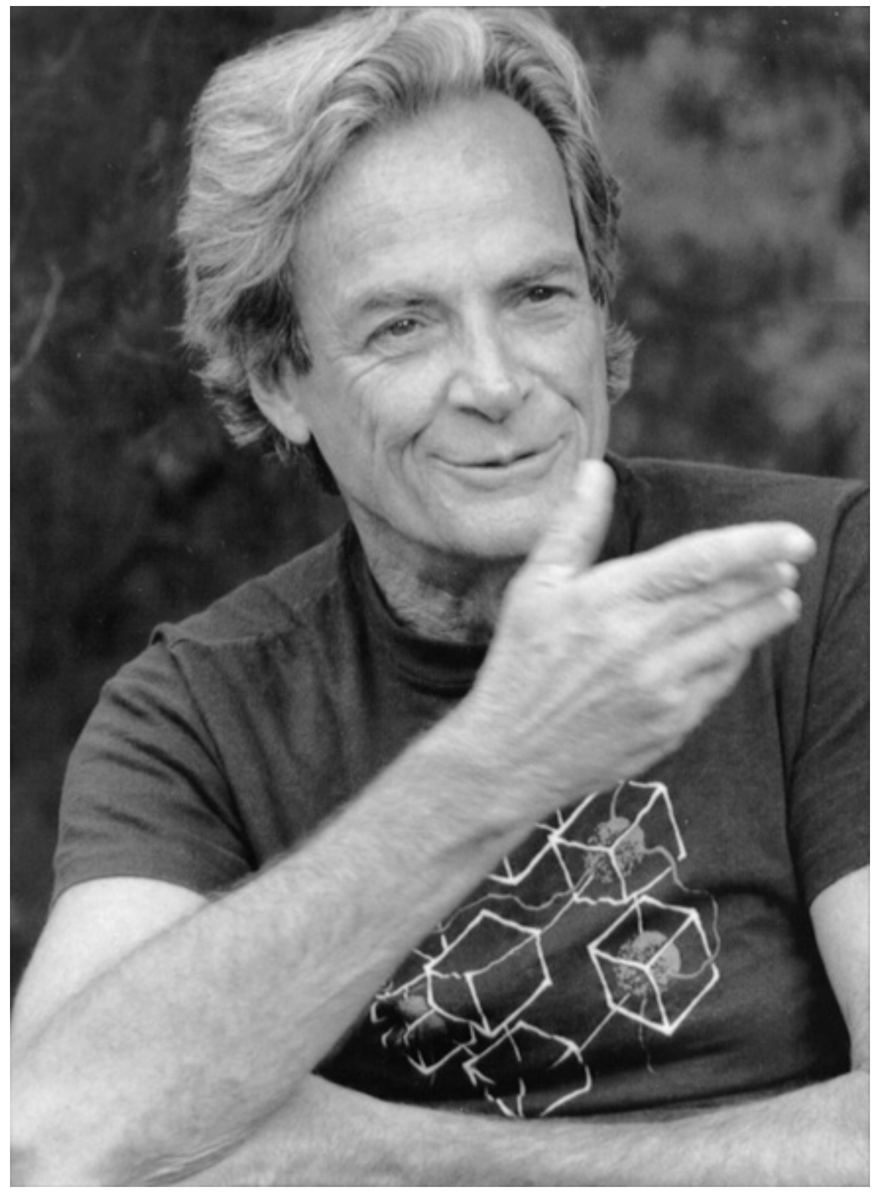
Discriminative learning

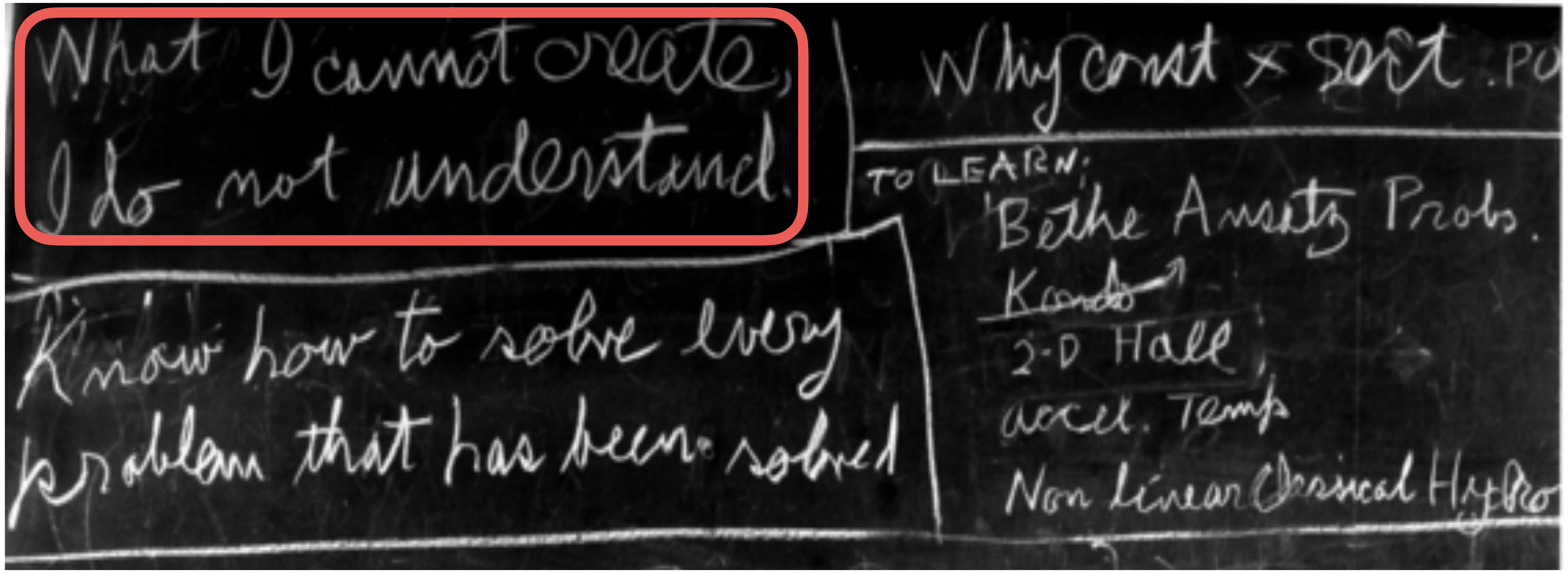
$$y = f(\boldsymbol{x})$$

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

Generative learning

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





ELSEVIER

Progress in Brain Research

Volume 165, 2007, Pages 535–547

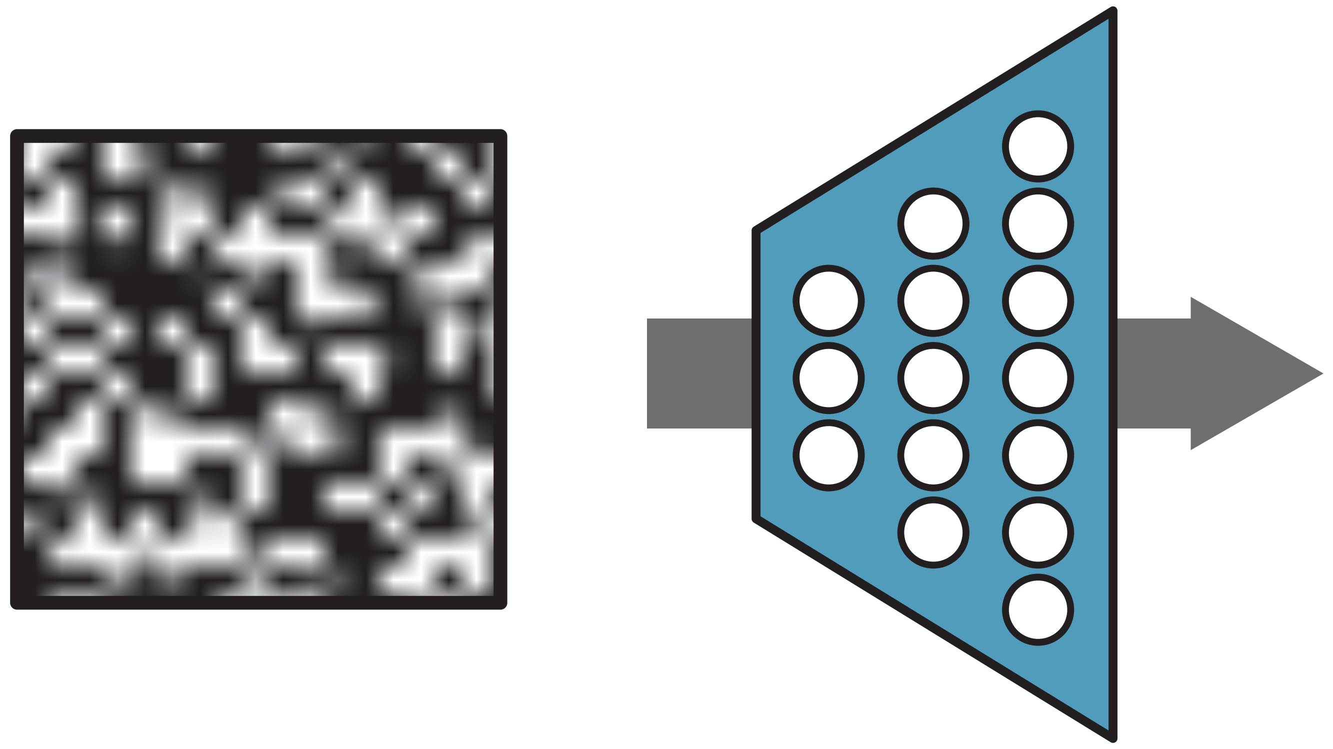
Computational Neuroscience: Theoretical Insights into Brain Function

To recognize shapes, first learn to generate images

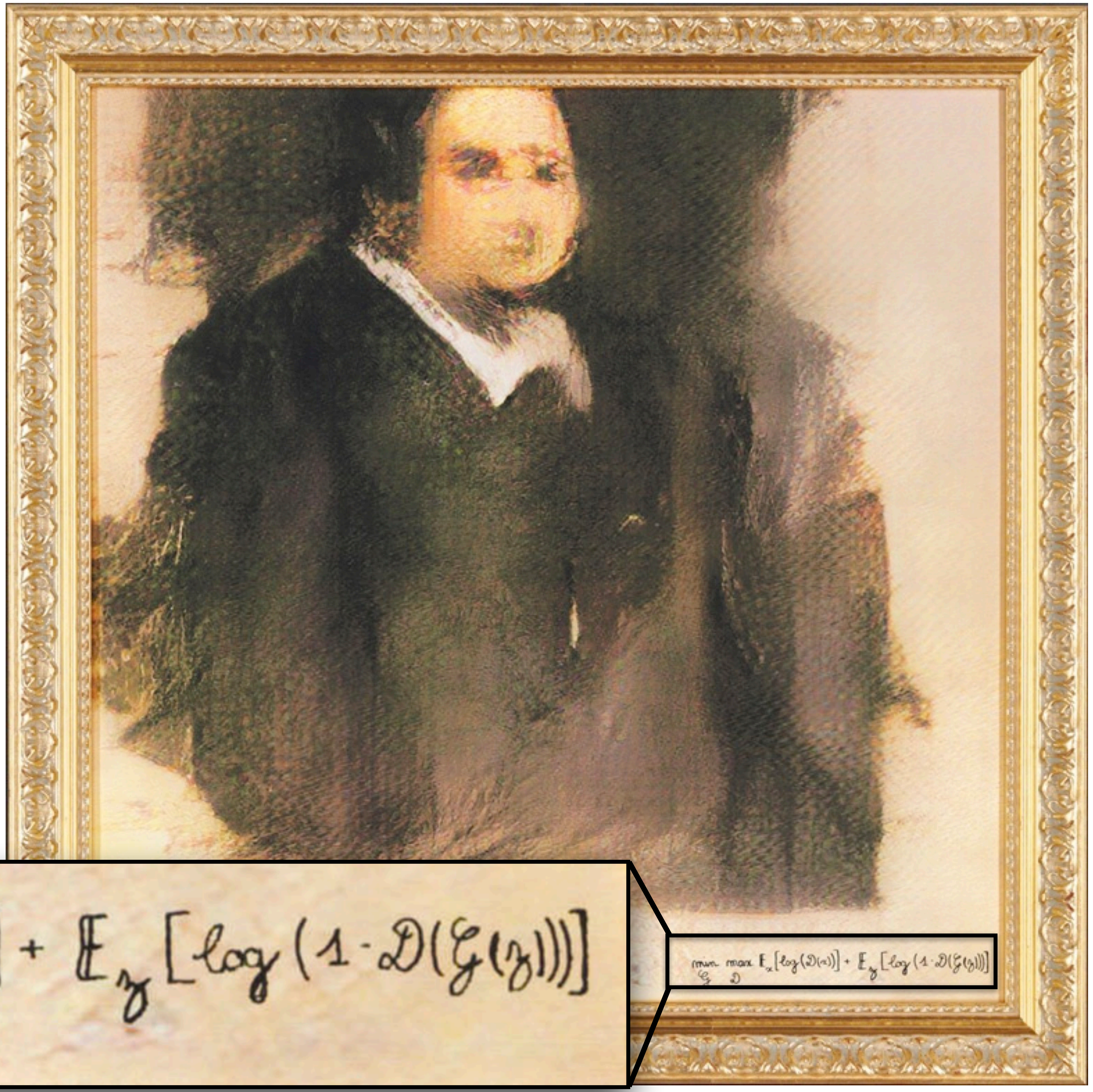
Geoffrey E. Hinton

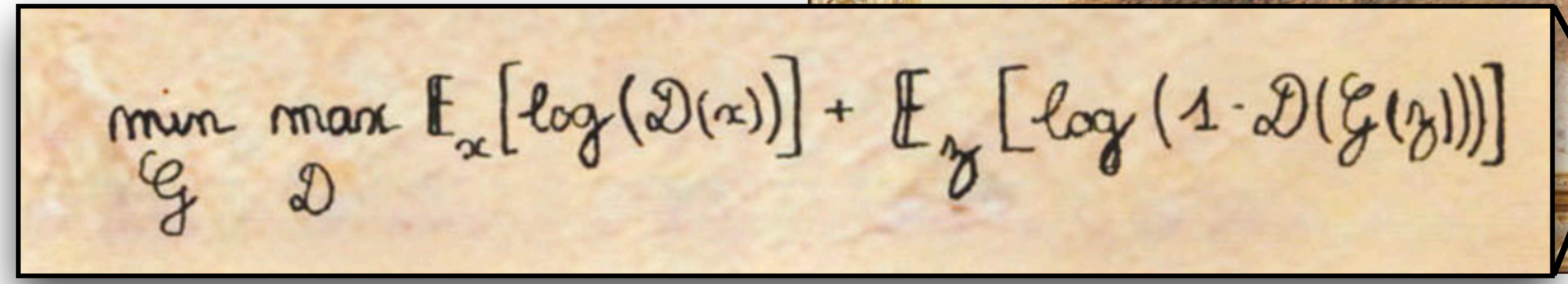
Department of Computer Science, University of Toronto, 10 Kings College Road, Toronto, M5S 3G4 Canada

Generated arts



Gaussian Noise Generative Network



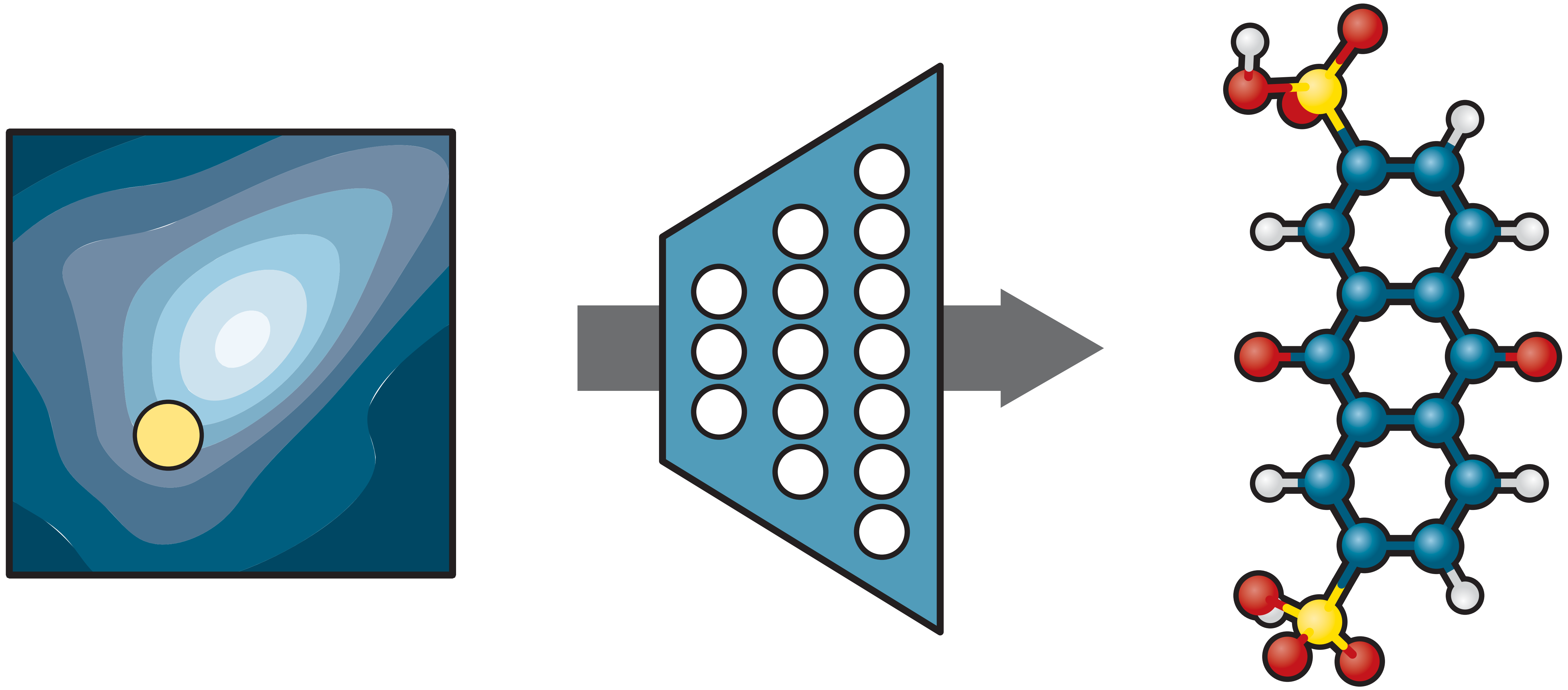


$432,500
25 October 2018
Christie's New York

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

Generating molecules

Latent attributes



Physical configurations

Simple Distributions

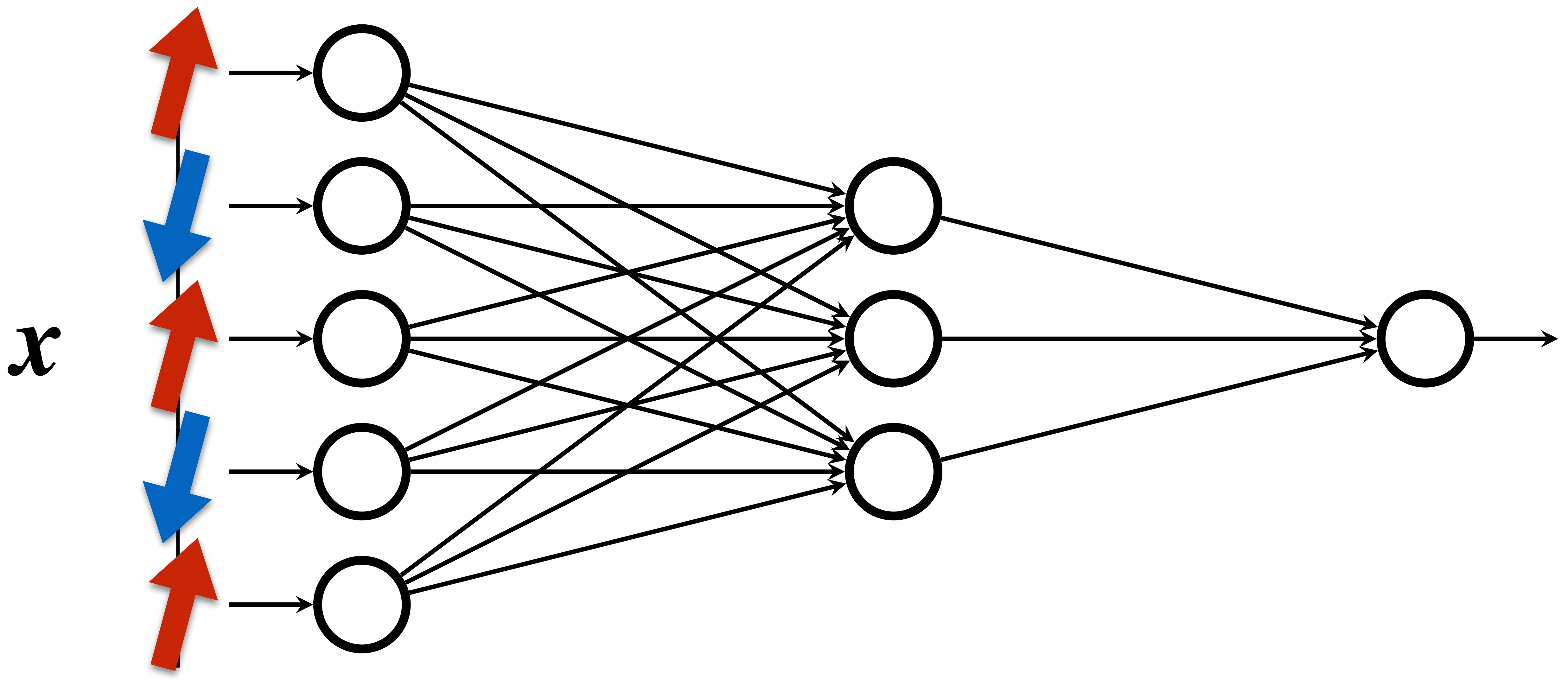
Generate

Inference

Complex Distribution

Sanchez-Lengeling & Aspuru-Guzik, Inverse molecular design using machine learning: Generative models for matter engineering, Science '18

So, what is the fuss ?



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

Normalization ?

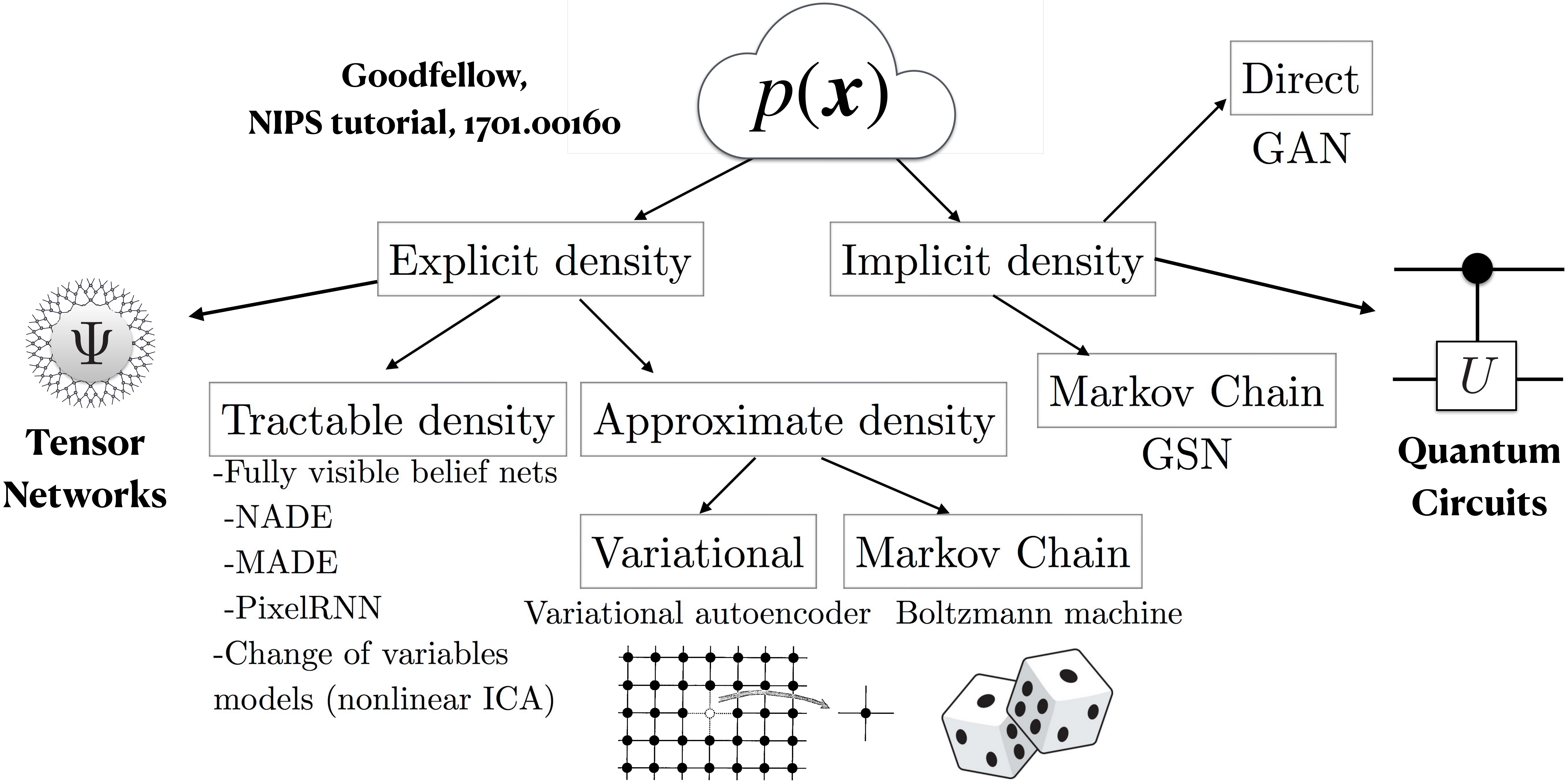
$$\int d\boldsymbol{x}\, p(\boldsymbol{x}) = 1$$

Sampling ?

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

Generative models and their physics genes

Goodfellow, NIPS tutorial, 1701.00160



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

Generative Models for Physicists

Lei Wang*

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

October 28, 2018

Abstract

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

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

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

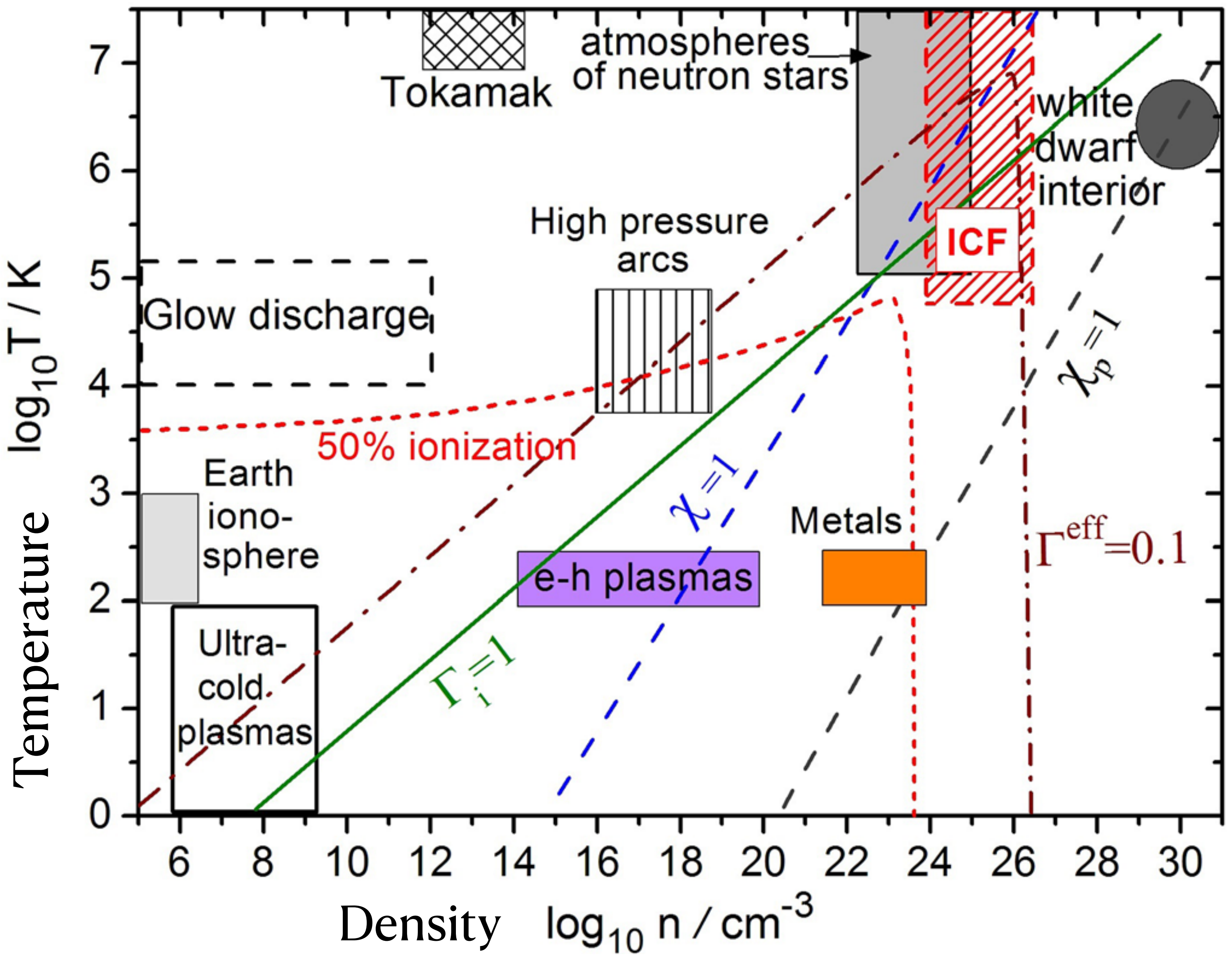
CONTENTS

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

Ab-initio study of quantum matters at finite T

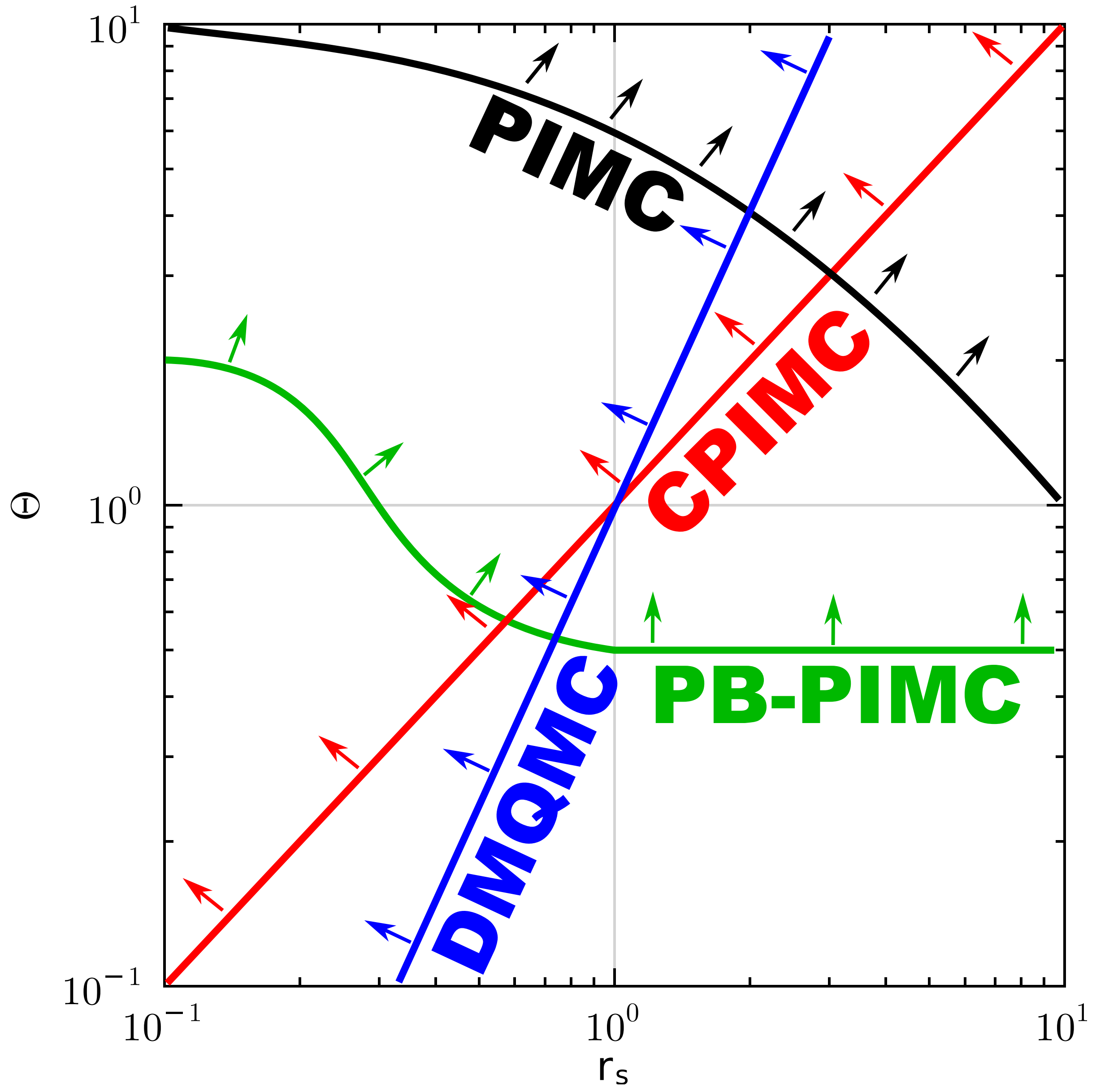
$$H = -\sum_i \frac{\hbar^2}{2m_e}\nabla_i^2 - \sum_{I,i}\frac{Z_I e^2}{|R_I - r_i|} + \frac{1}{2}\sum_{i\neq j}\frac{e^2}{|r_i - r_j|} - \sum_I \frac{\hbar^2}{2m_I}\nabla_I^2 + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J e^2}{|R_I - R_J|}$$

$$Z = \mathrm{Tr}(e^{-\beta H})$$



Bonitz et al, Phys. Plasmas '20

Application range of the workhorse quantum Monte Carlo



Dornheim et al, Phys. Plasmas '17

The classical variational free-energy approach

Gibbs–Bogolyubov-Feynman variational principle

$$F = \int dx\, p(\boldsymbol{x}) \left[k_B T \ln p(\boldsymbol{x}) + H(\boldsymbol{x}) \right] \geq -k_B T \ln Z$$

entropy

energy

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

Richard P. Feynman

deep generative models !

[1]transcript of Professor Feynman's talk in 1987

Generative modeling

Known: samples

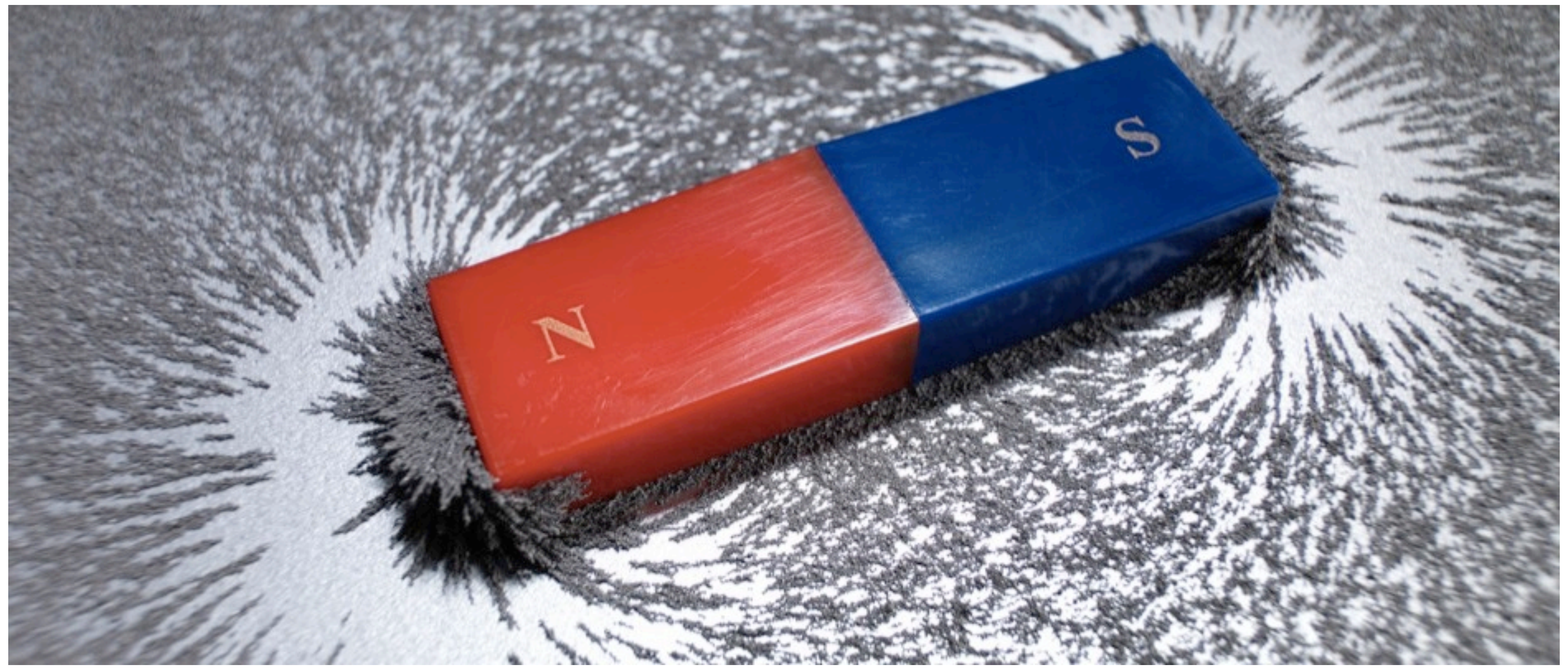
Unknown: generating distribution

Maximum likelihood estimation

"learn from data"

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

Statistical physics



Known: energy function

Unknown: samples, partition function

Variational free energy

"learn from Hamiltonian"

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

Deep variational free-energy approach

Use deep generative models as the variational density

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

$$\underbrace{k_B T \ln p(\boldsymbol{x})}_{\downarrow \text{ entropy}} \qquad \underbrace{H(\boldsymbol{x})}_{\downarrow \text{ energy}}$$

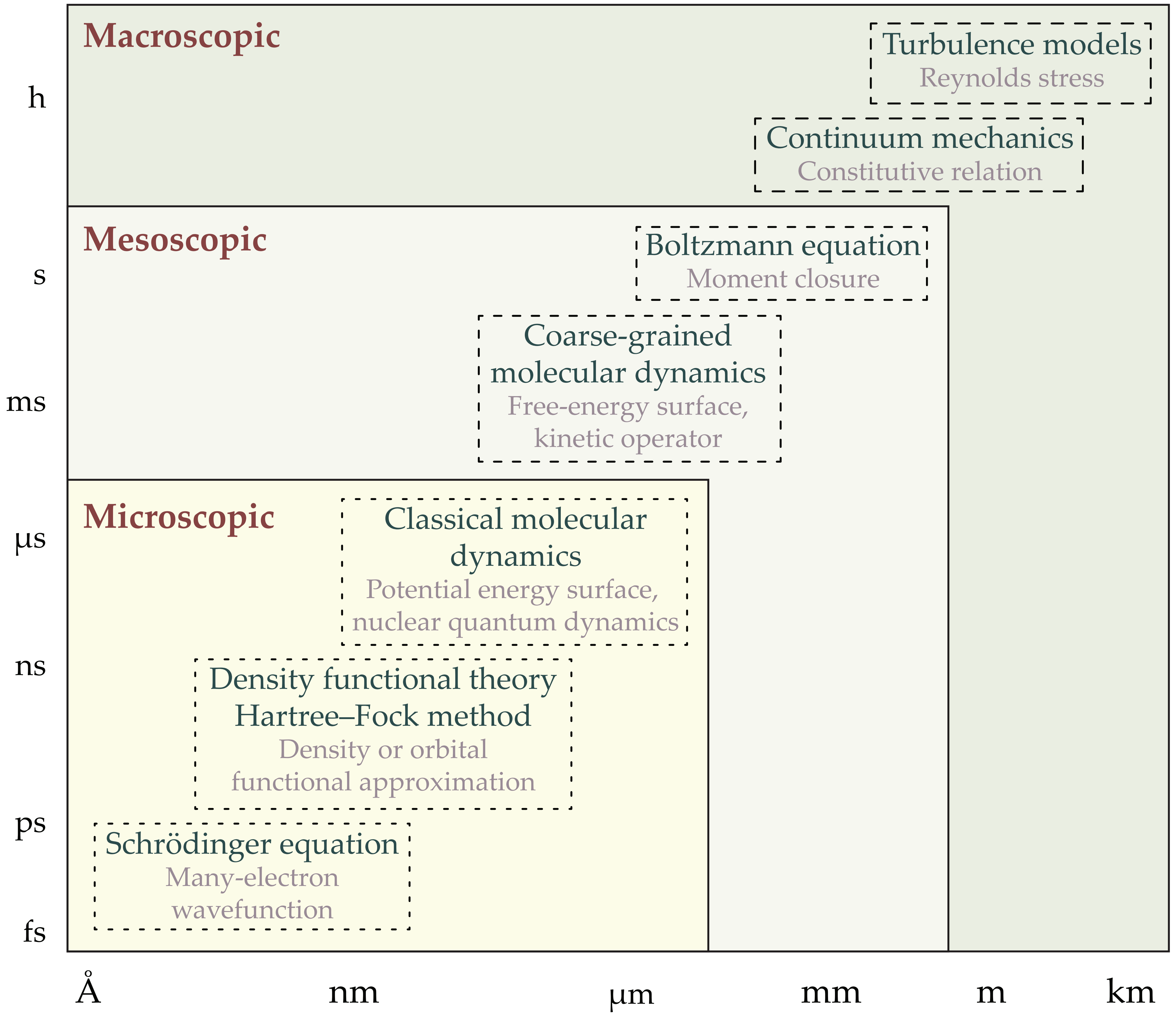
😁 entropy energy

✓ Tractable entropy ✓ Direct sampling

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

Variational free-energy in the context

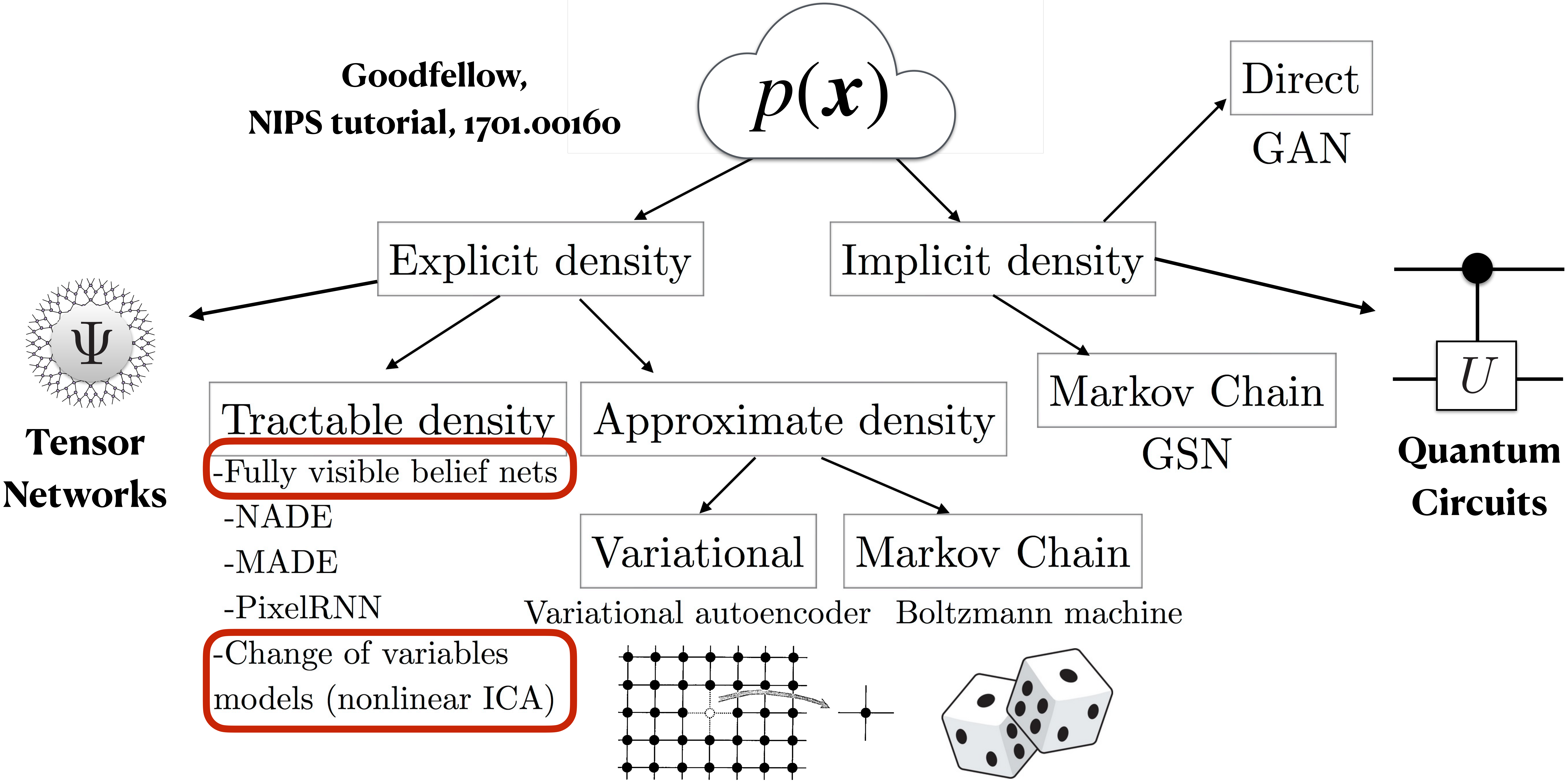
E, Han,Zhang, Physics Today 2020



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

more fundamental, more difficult, more limited

Generative models and their physics genes



Generative modeling with normalizing flows

WaveNet 1609.03499 1711.10433

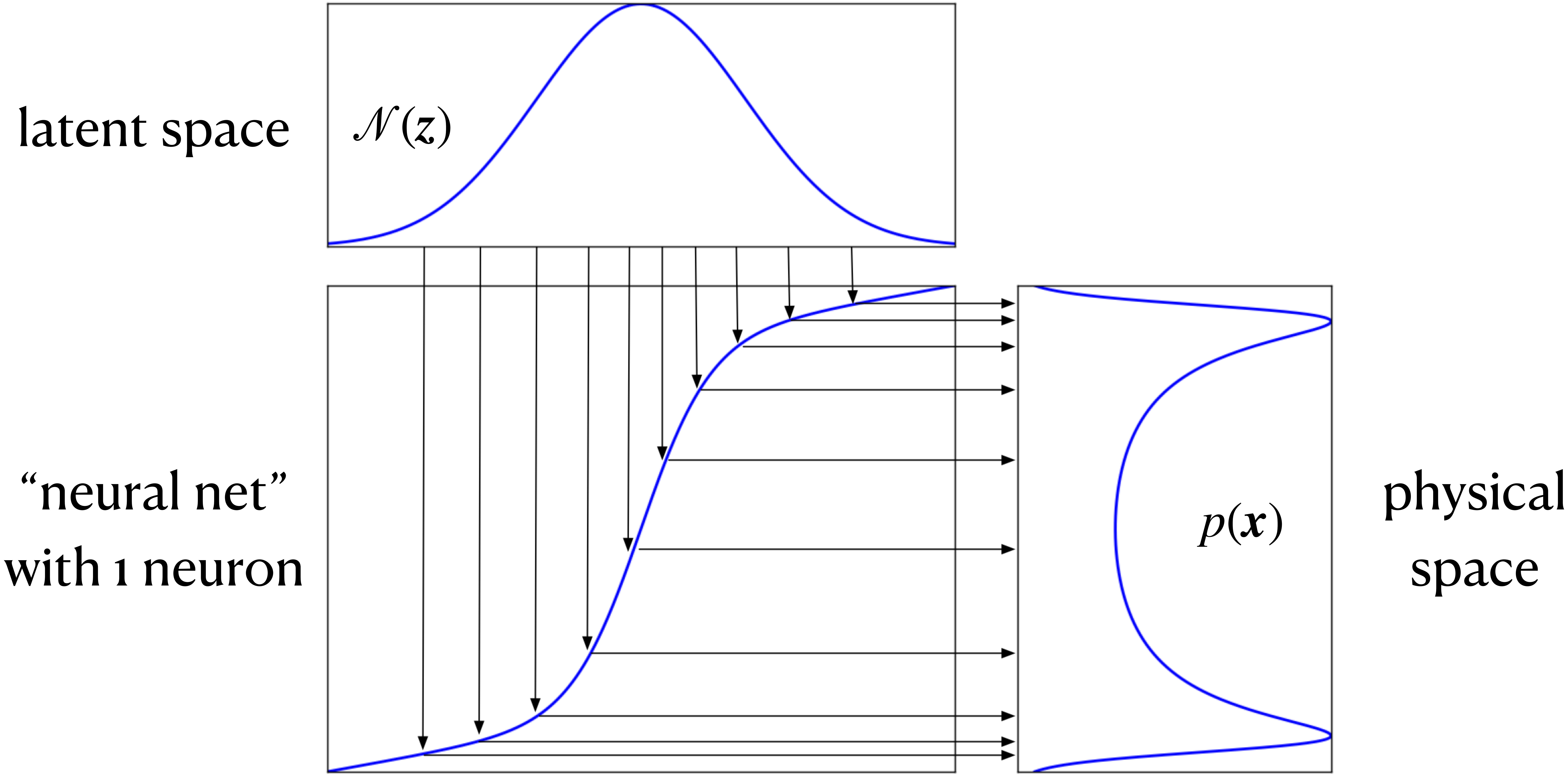
https://deepmind.com/blog/wavenet-generative-model-raw-audio/

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

Glow 1807.03039

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

Normalizing flow in a nutshell



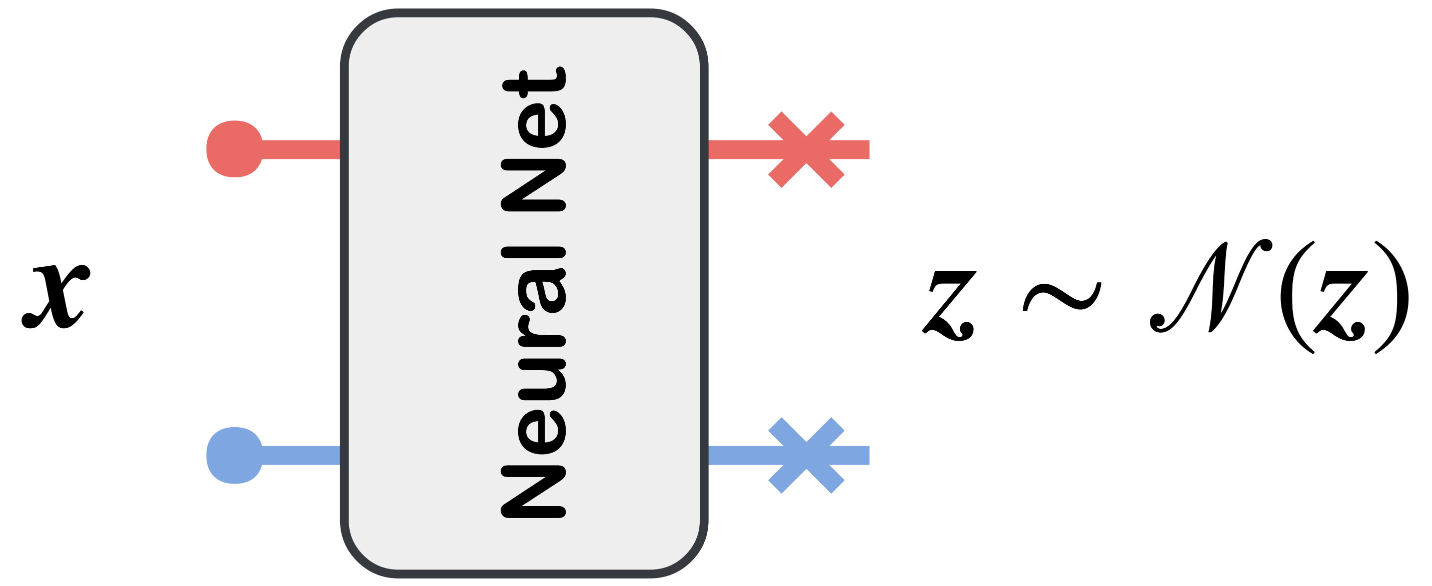
Normalizing Flows

Change of variables $x \leftrightarrow z$ **with deep neural nets**

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

Review article 1912.02762
Tutorial https://iclr.cc/virtual_2020/speaker_4.html

composable, differentiable, and invertible mapping between manifolds

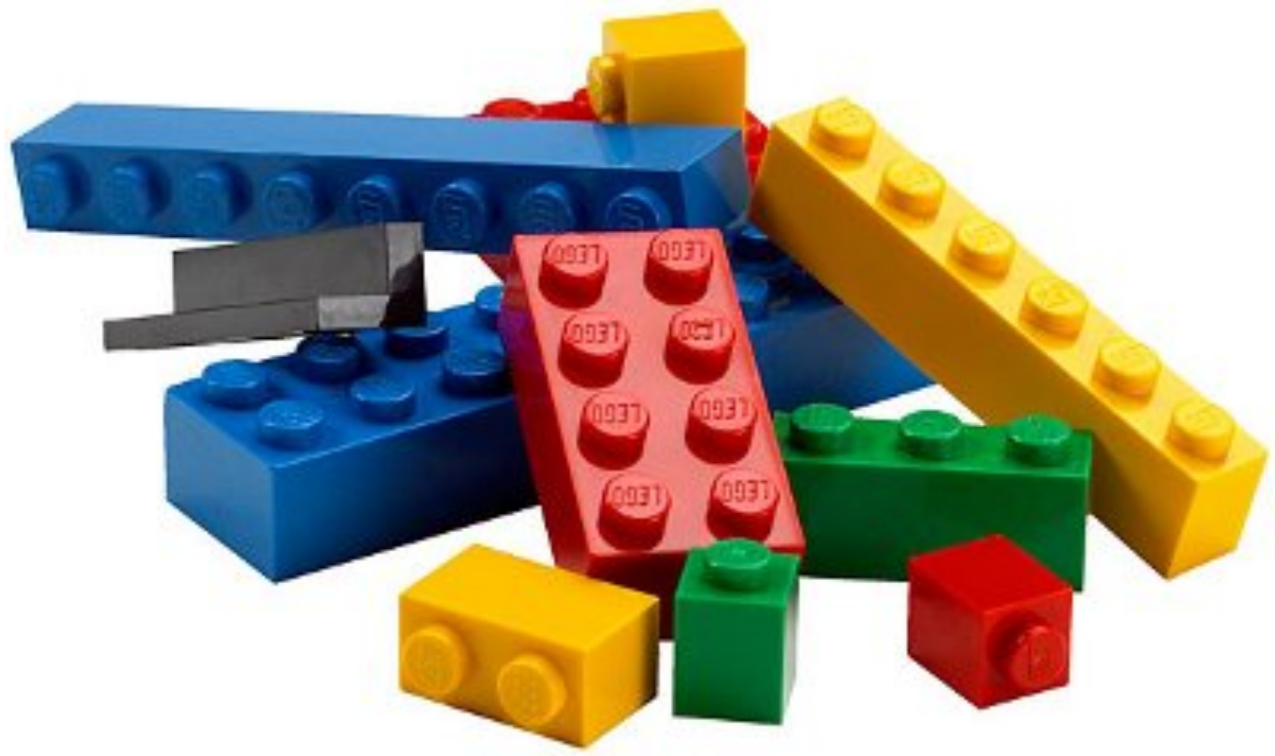


Learn probability transformations with normalizing flows

Got this name in Tabak & Vanden-Eijnden, Commun. Math. Sci. '10

Architecture design principle

Composability

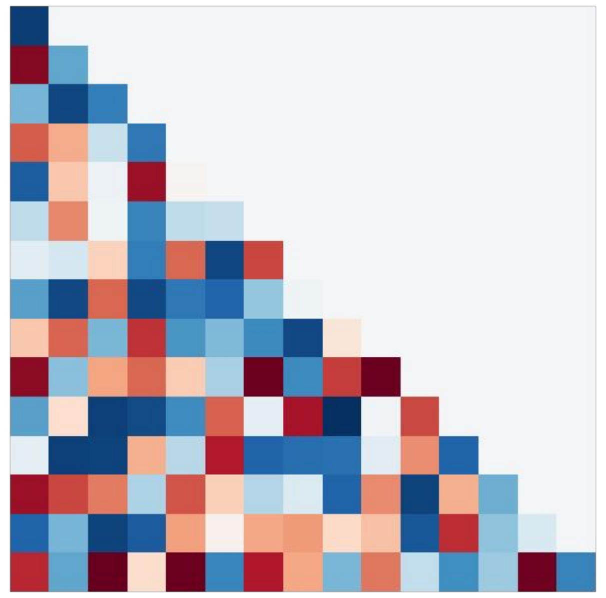


$$z = \mathscr{T}(\boldsymbol{x})$$

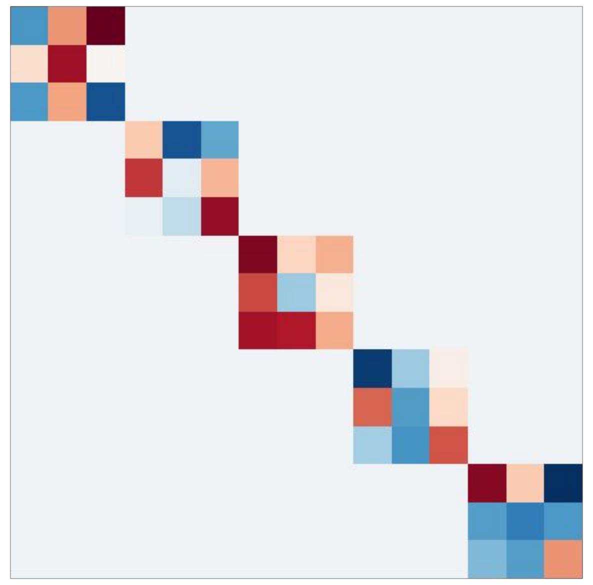
$$\mathscr{T} = \mathscr{T}_1 \circ \mathscr{T}_2 \circ \mathscr{T}_3 \circ \cdots$$

Balanced efficiency & inductive bias

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



Autoregressive



Neural RG

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

Continuous flow

Example of a building block

Forward

arbitrary neural nets

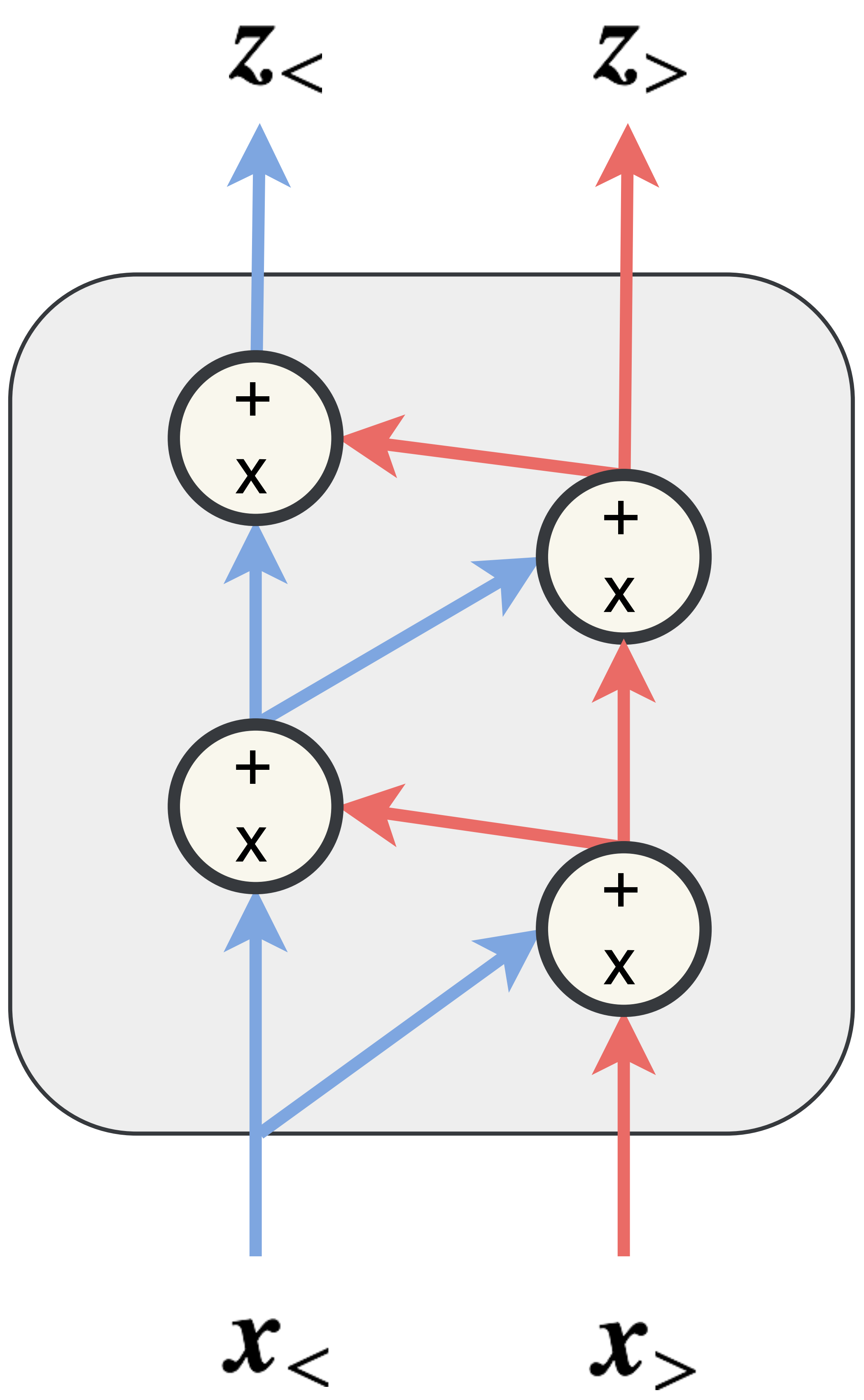
$$\begin{cases} x_< = z_< \\ x_> = z_> \odot e^{s(z_<)} + t(z_<) \end{cases}$$

Inverse

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

Log-Abs-Jacobian-Det

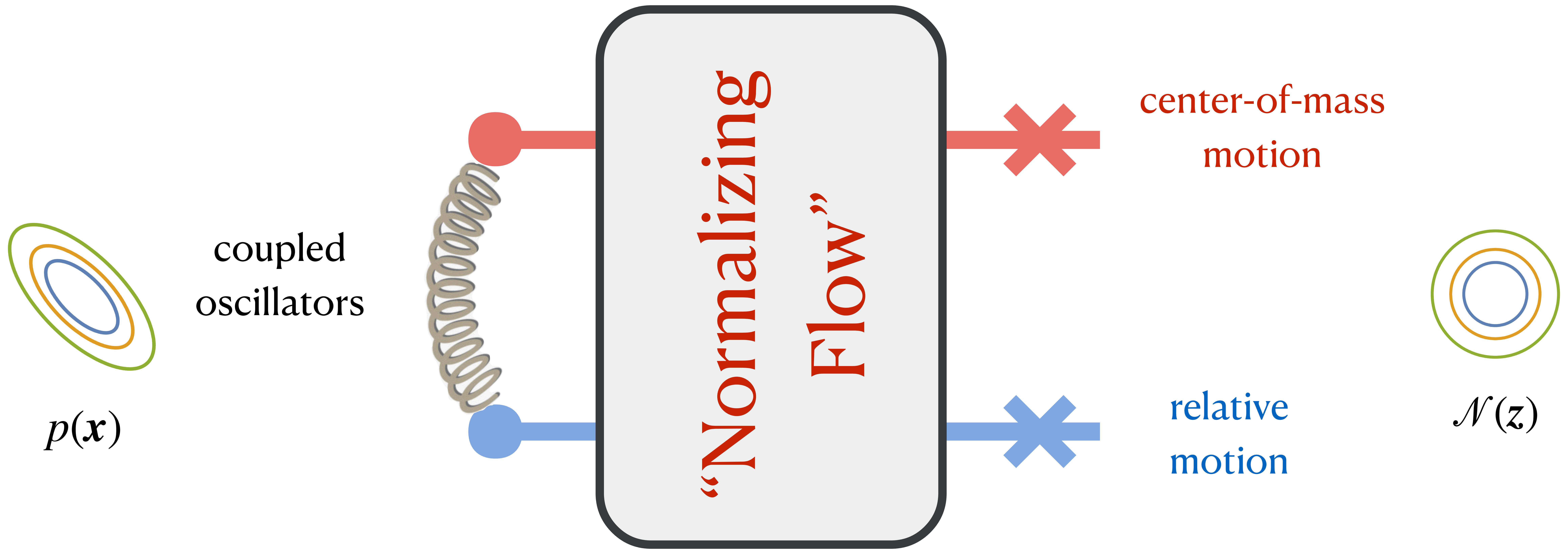
$$\ln \left| \det \left(\frac{\partial x}{\partial z} \right) \right| = \sum_i [s(z_<)]_i$$



Real NVP, Dinh et al,1605.08803

Turns out to have surprising connection Störmer–Verlet integration (later)

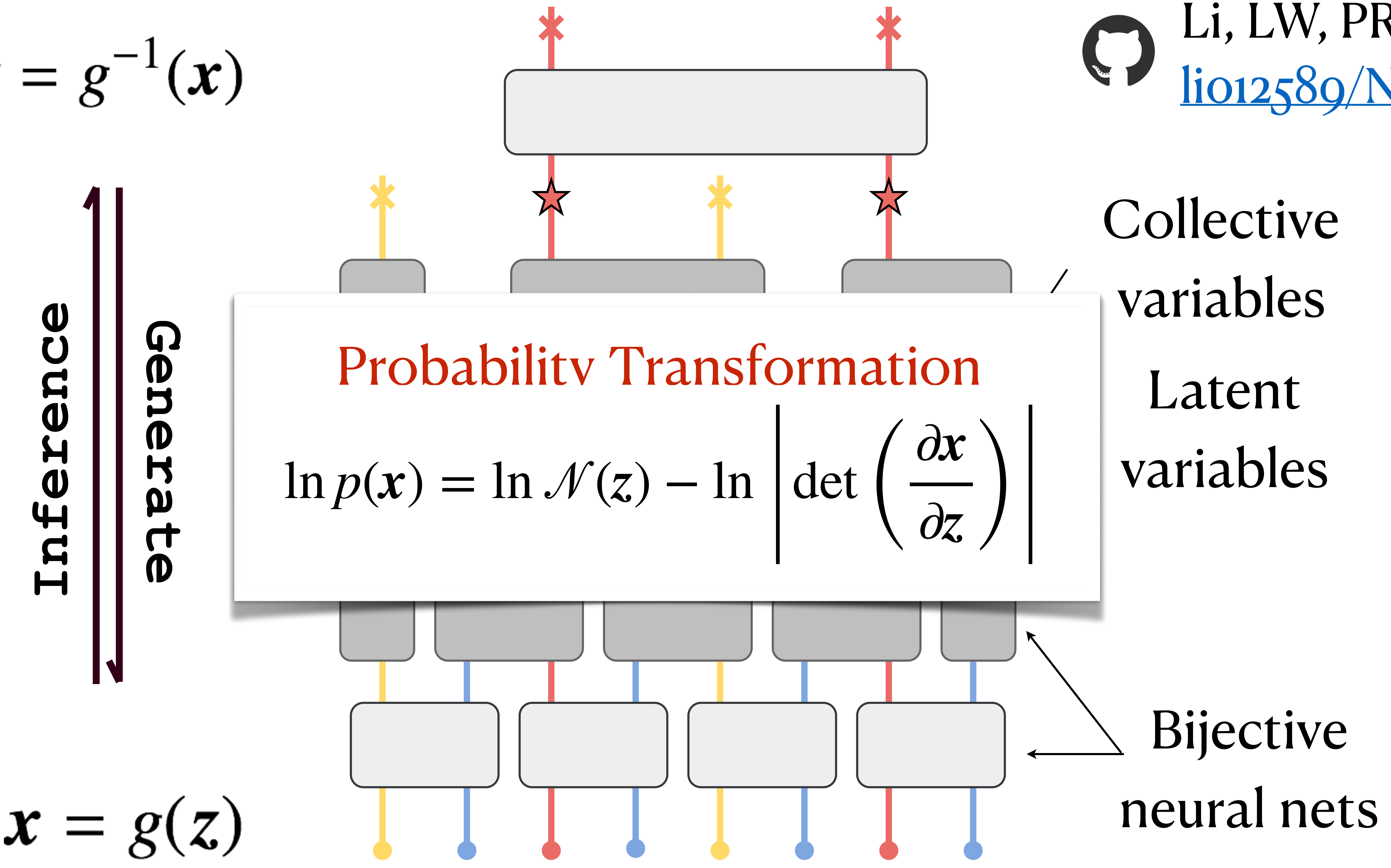
Normalizing flow in physics



Neural Network Renormalization Group

Li, LW, PRL '18
li012589/NeuralRG

$$z = g^{-1}(x)$$



Probability Transformation

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

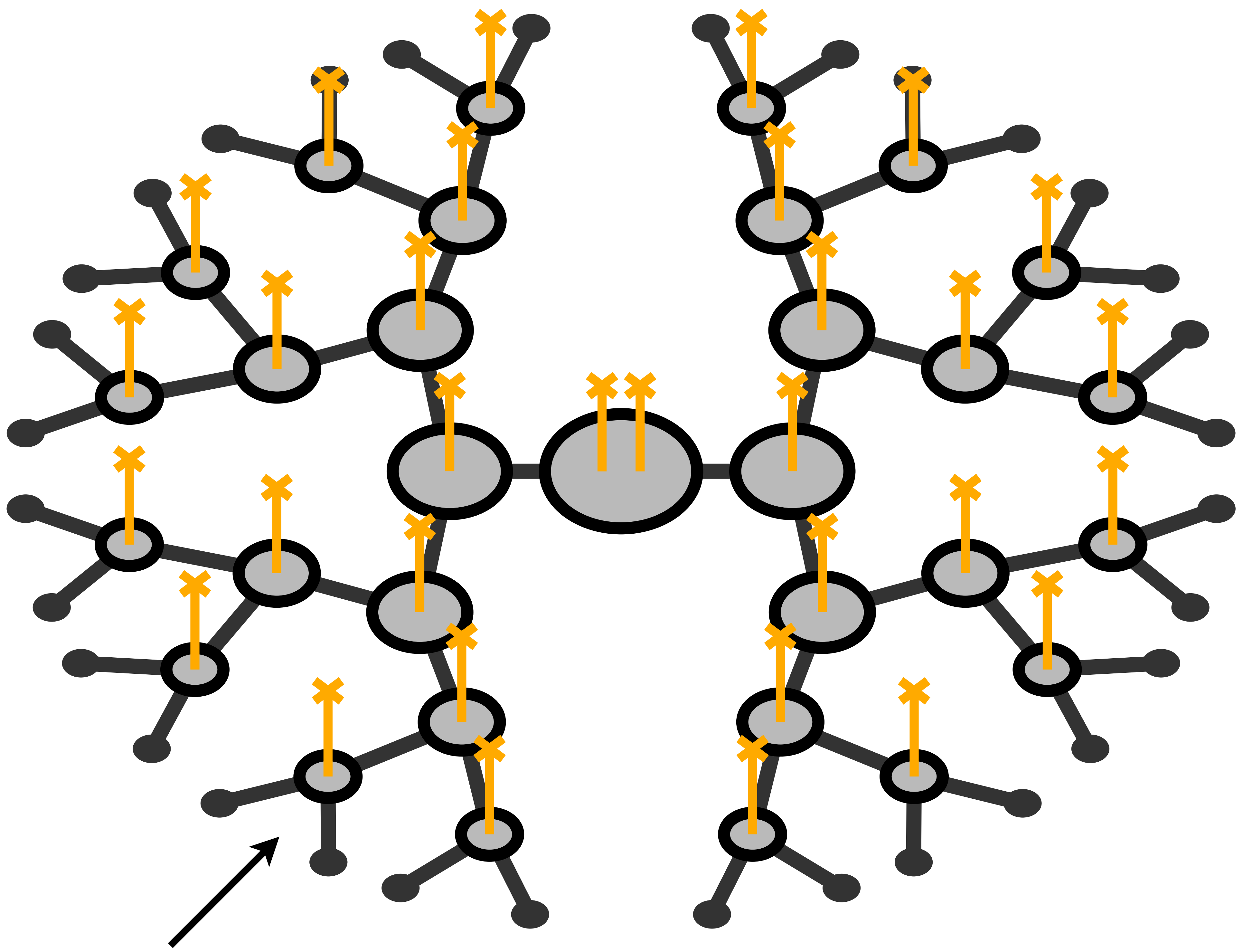
$$x = g(z)$$

Correlated classical variables

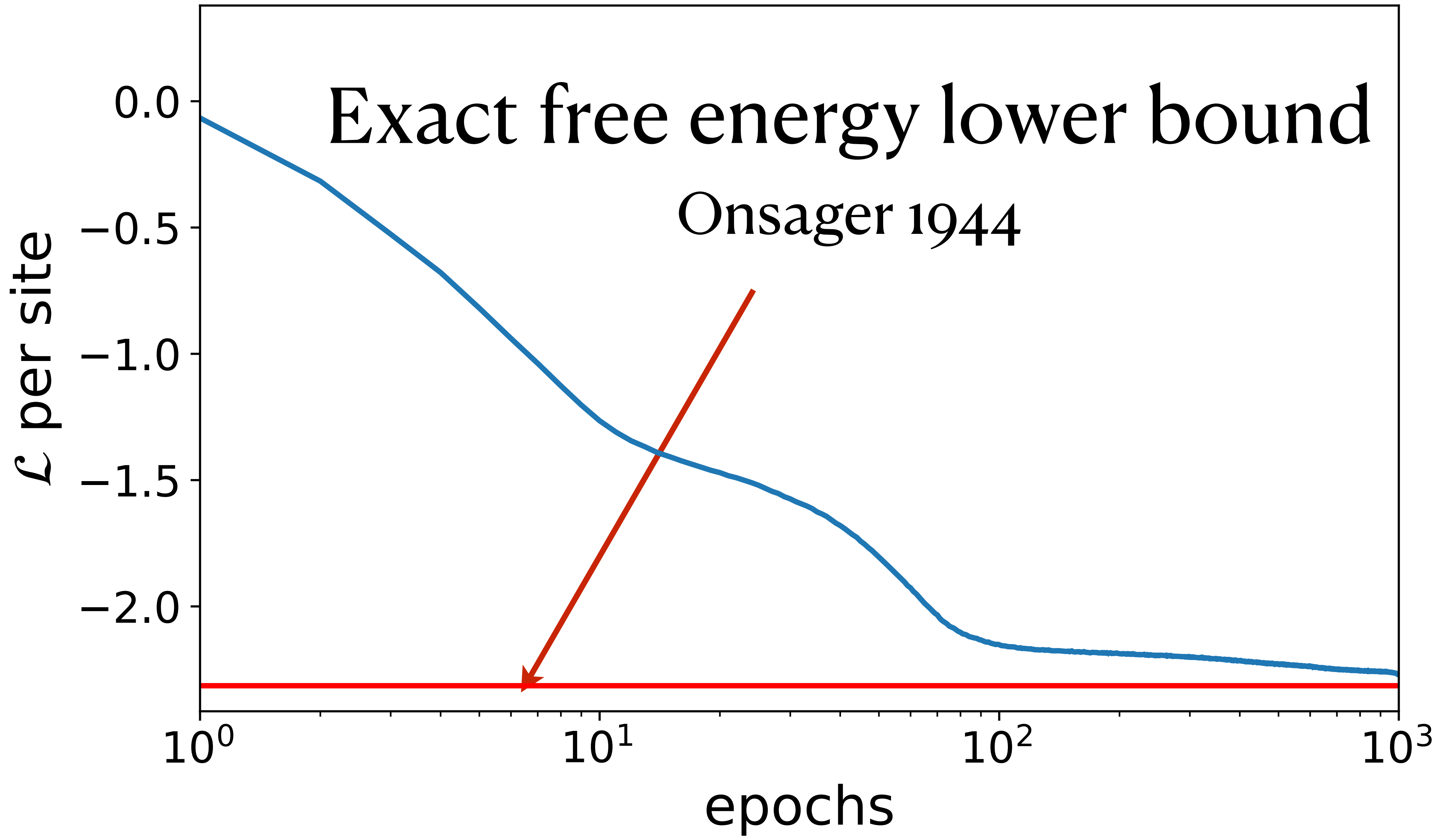
Neural network renormalization group

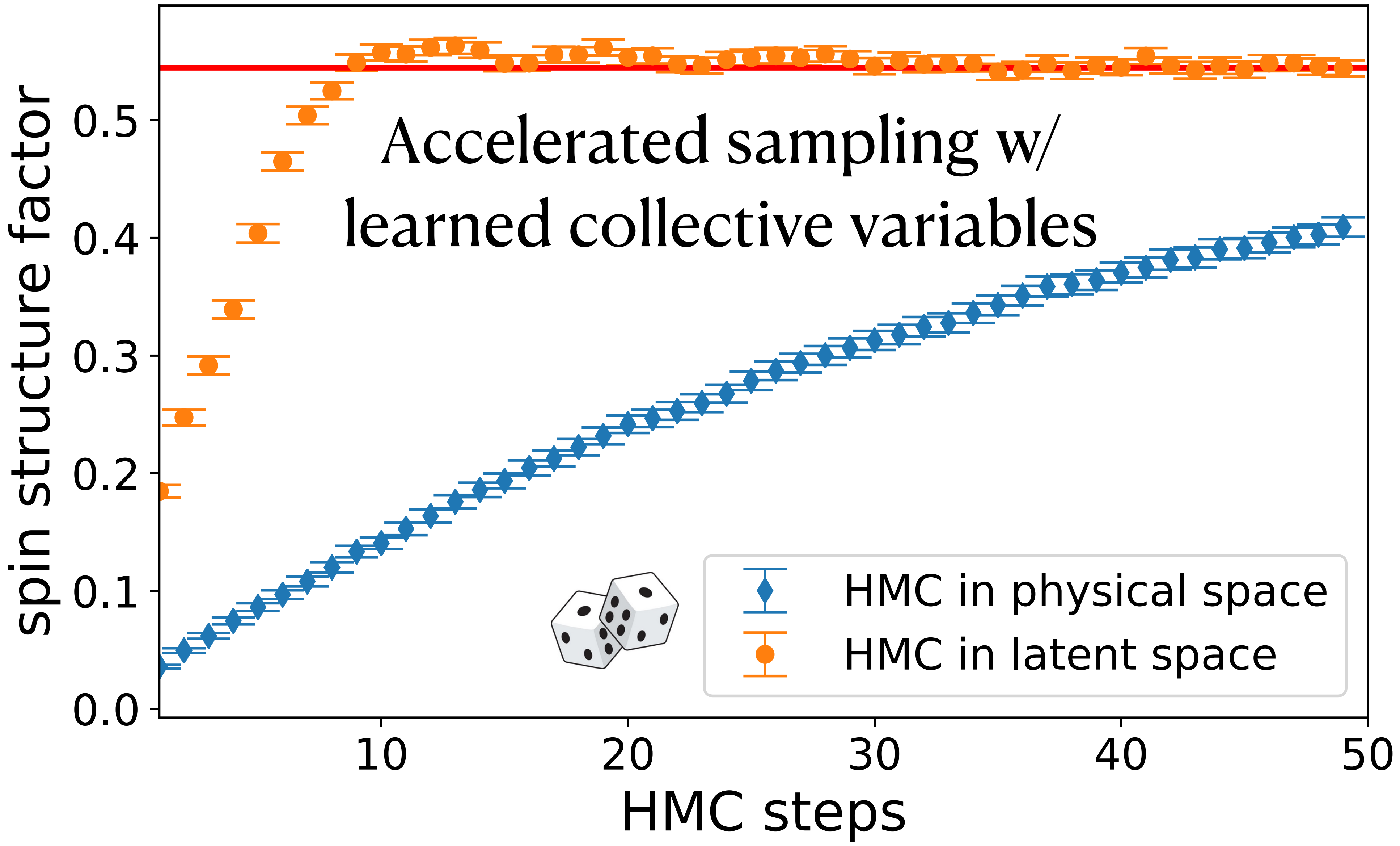
Li, LW, PRL '18

github.com/li012589/NeuralRG

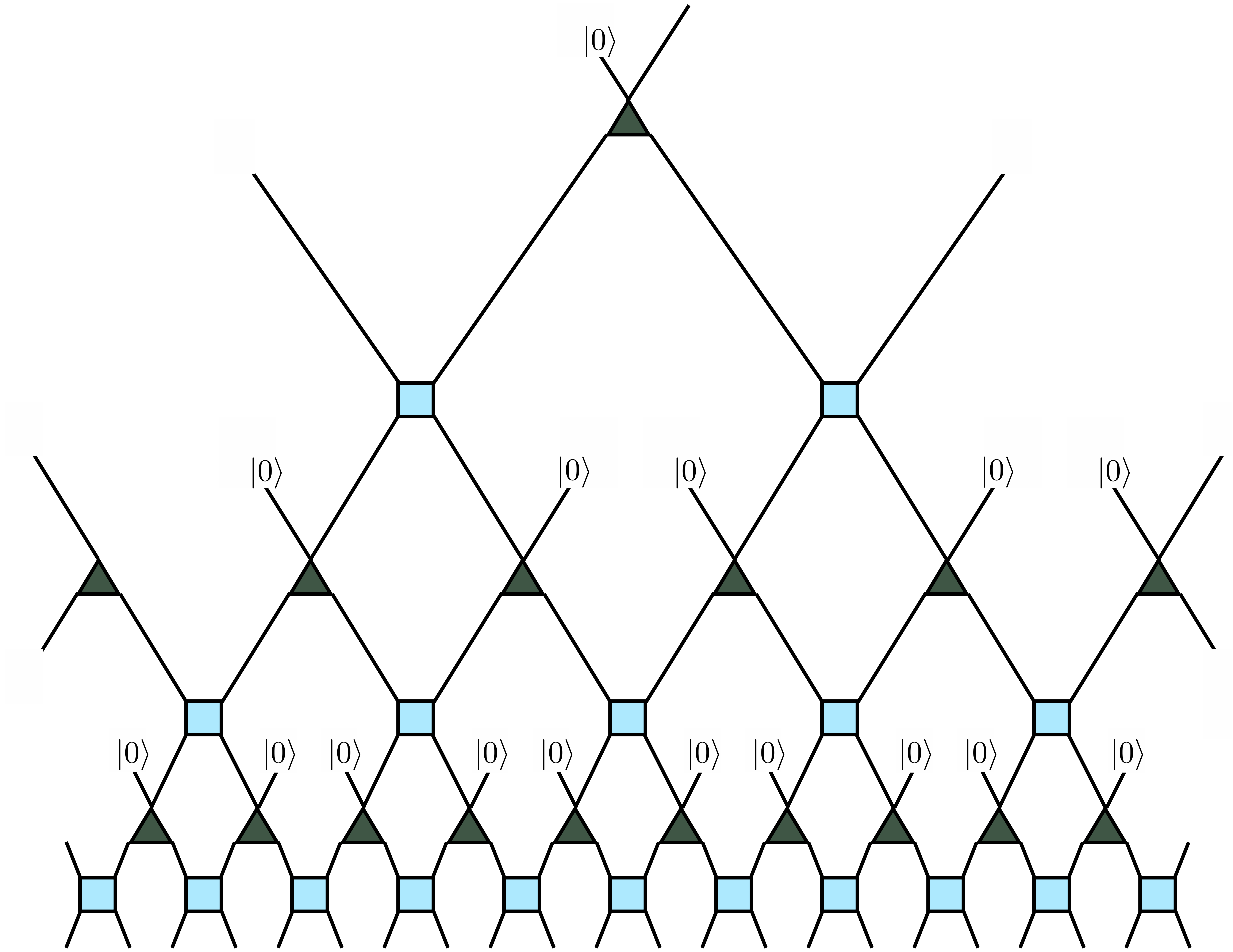


Bijective neural net

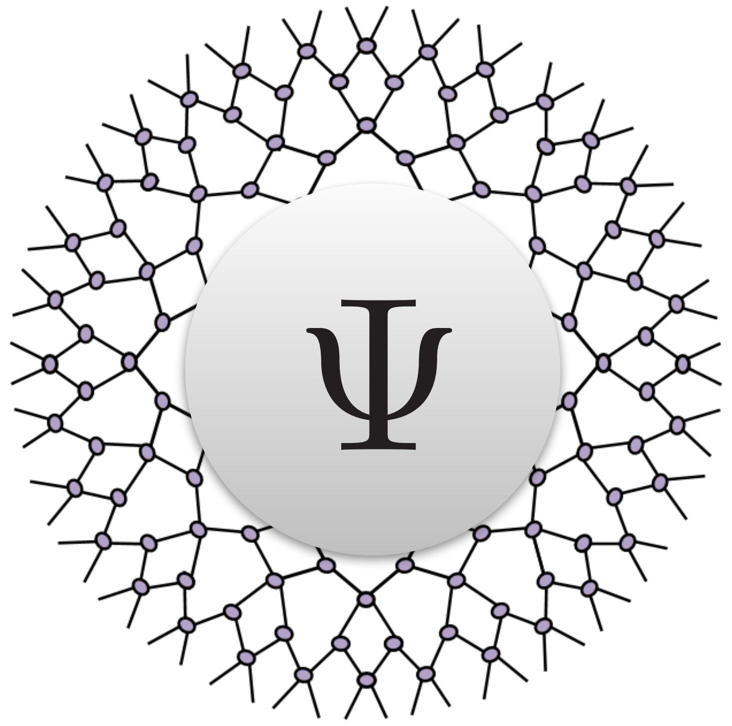




Quantum origin of the architecture

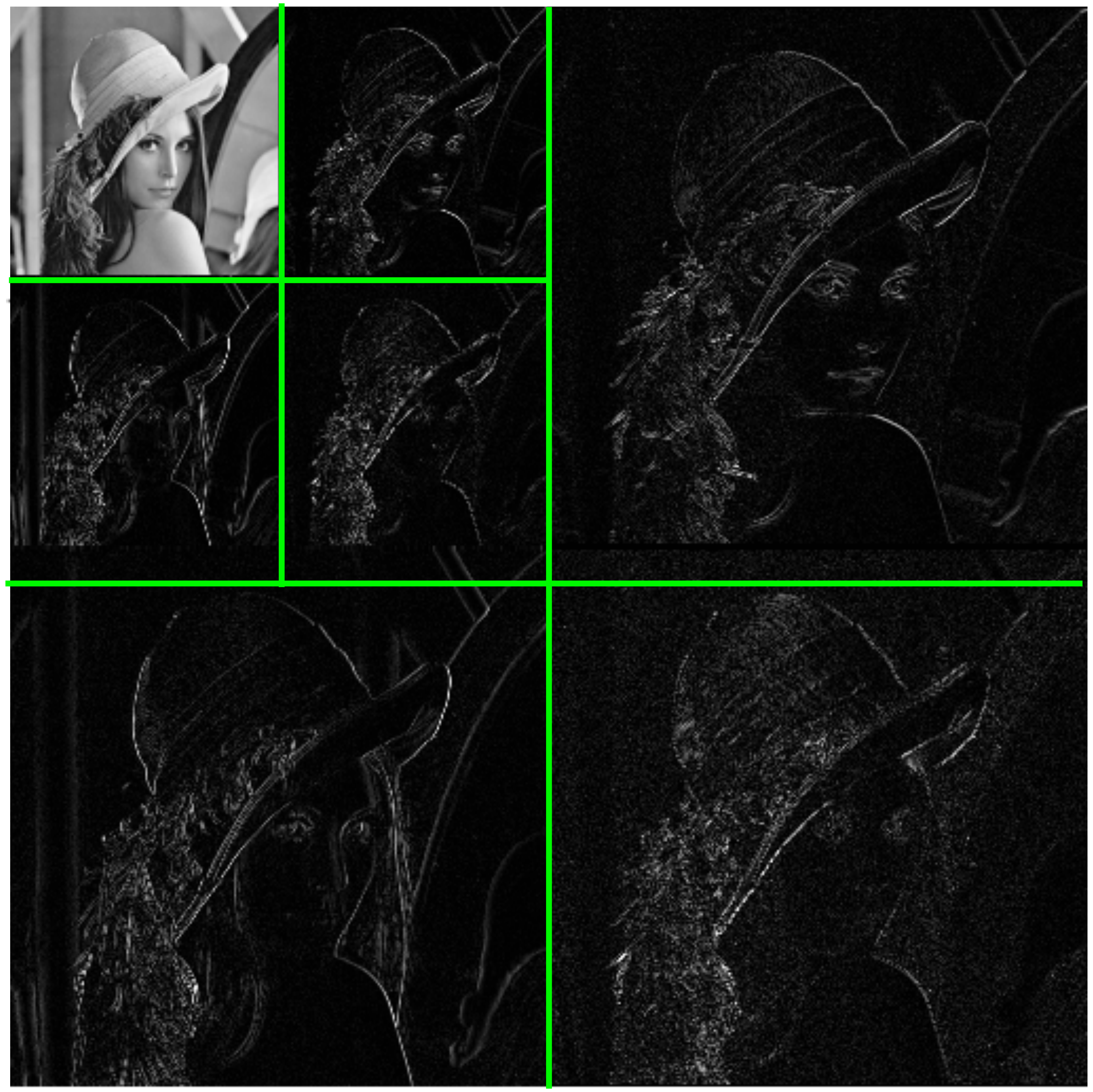


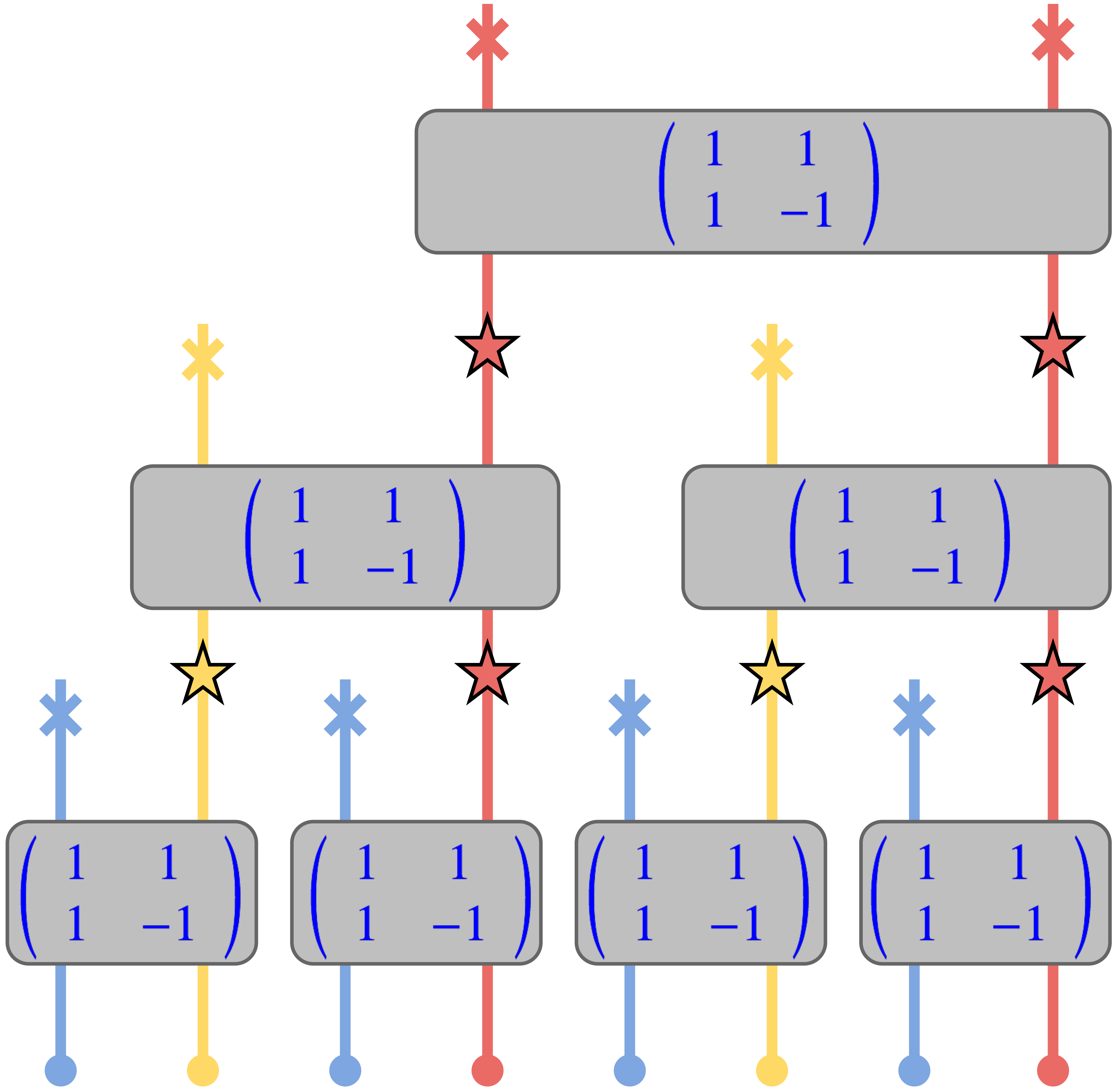
Entangled qubits



Multi-Scale
Entanglement
Renormalization
Ansatz

Connection to wavelets



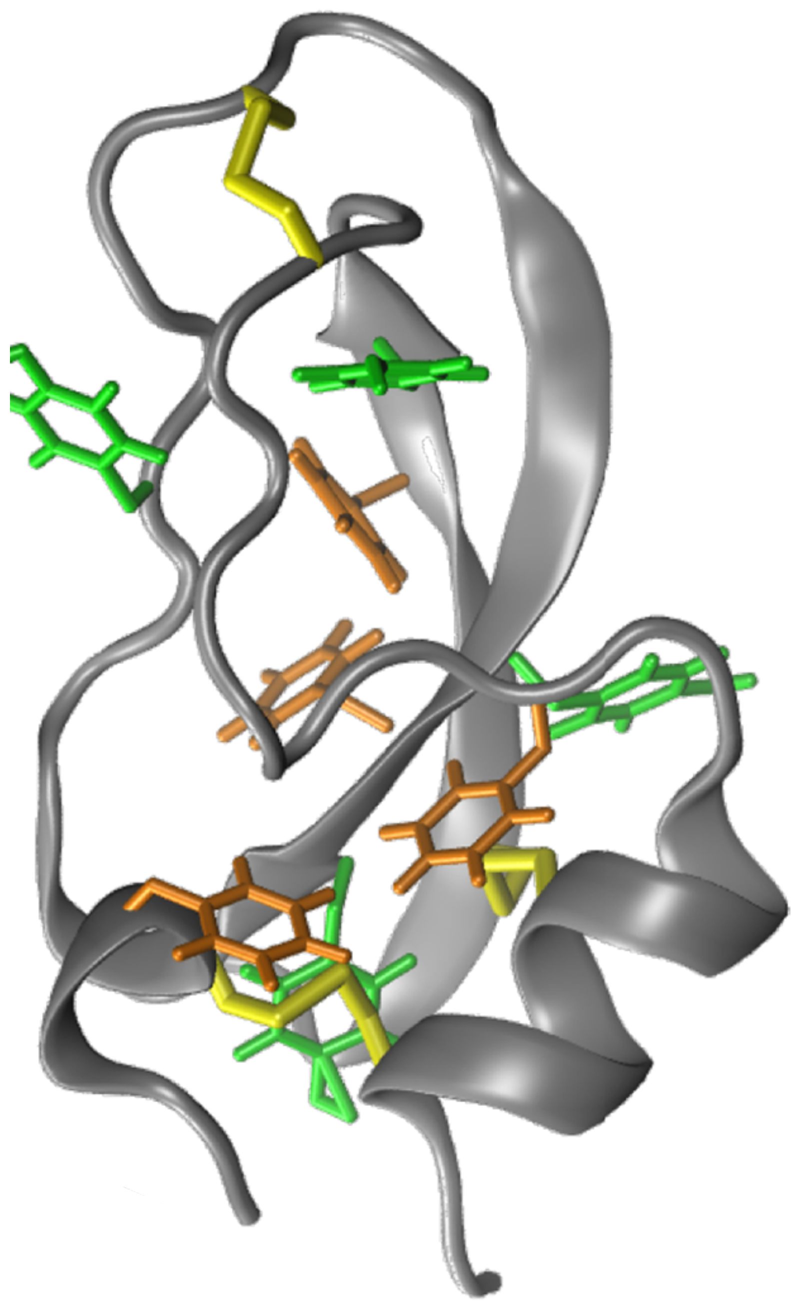


Nonlinear & adaptive generalizations of wavelets

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

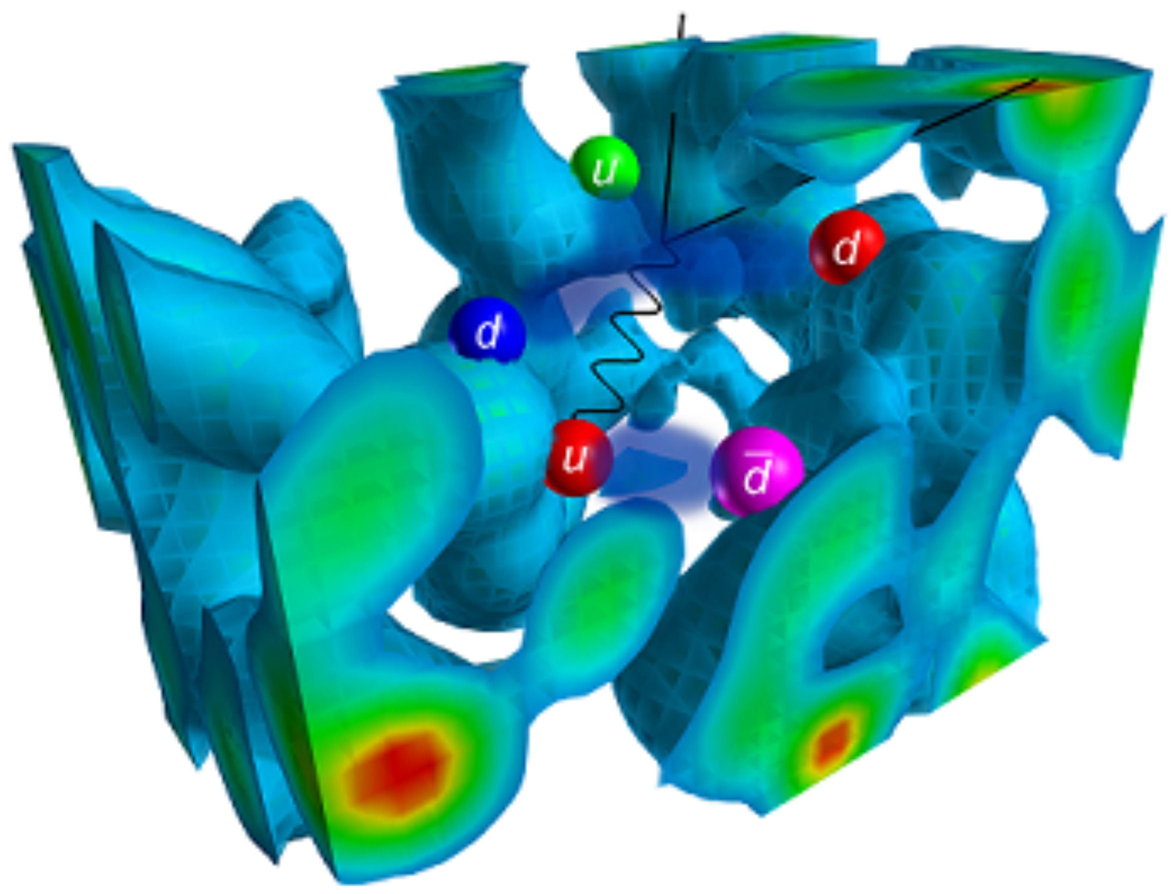
Normalizing flow in physics

Molecular simulation



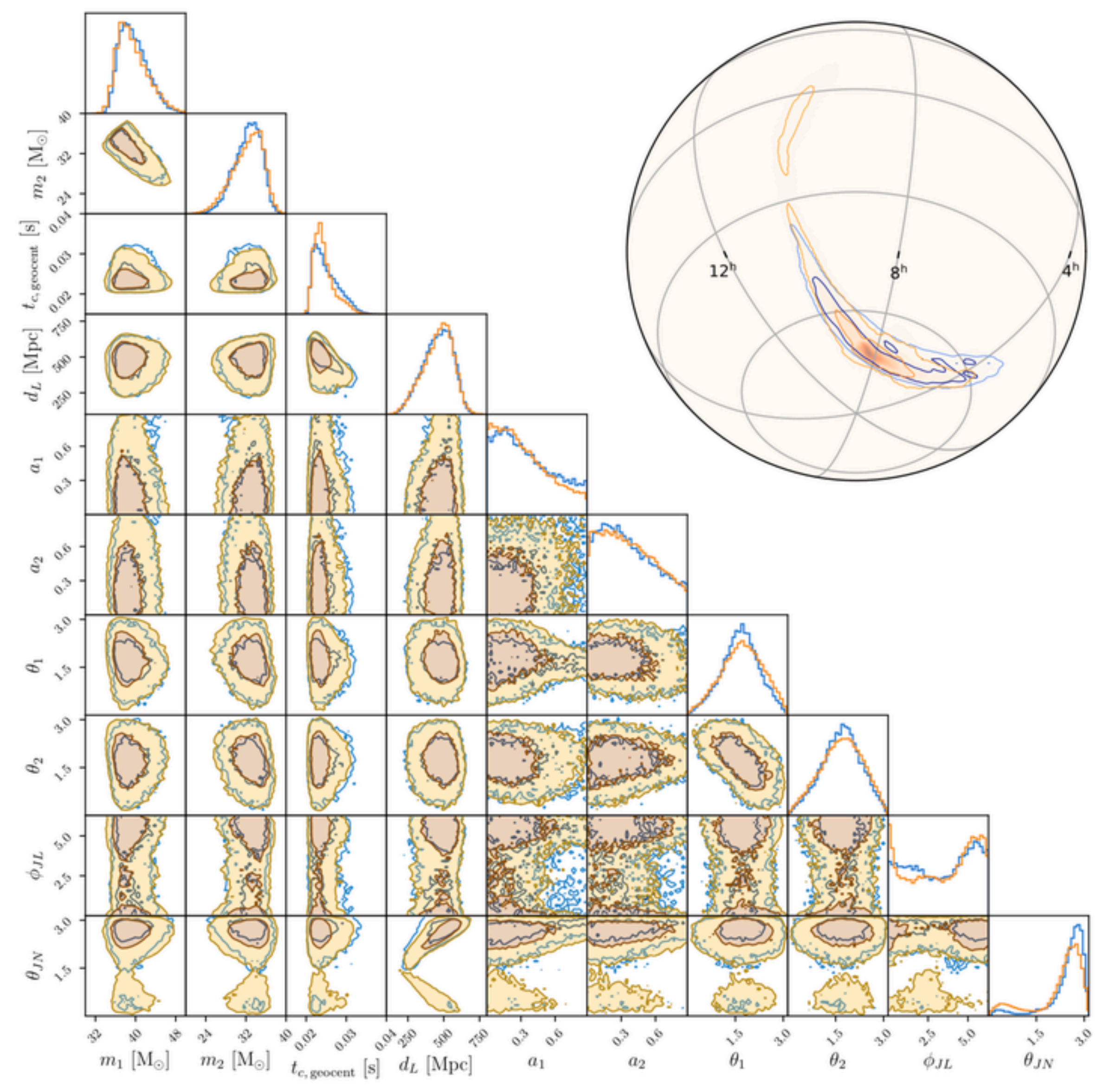
Noe et al, Science '19
Wirnsberger et al, JCP '20

Lattice field theory



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

Gravitational wave detection



Green et al, MLST '21
Dex et al, PRL '21

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

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

$\varepsilon \to 0$

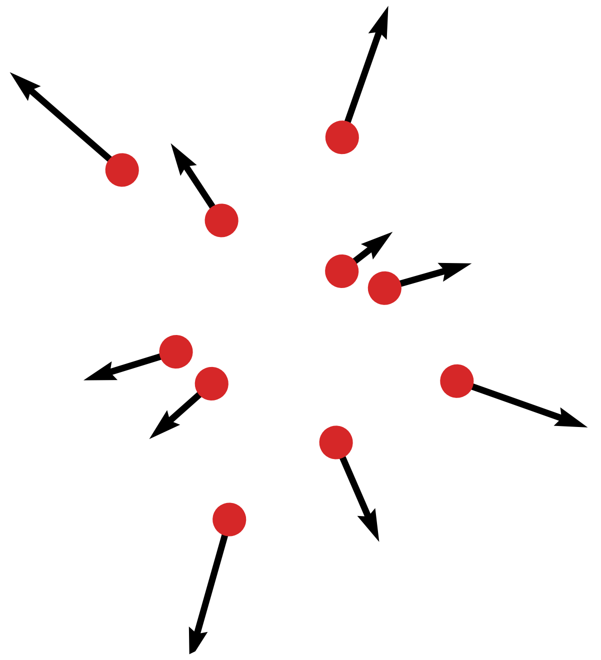
$t = T$ $t = 0$

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

Fluid physics behind flows

Zhang, E, LW 1809.10188
wangleiphy/MongeAmpereFlow

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

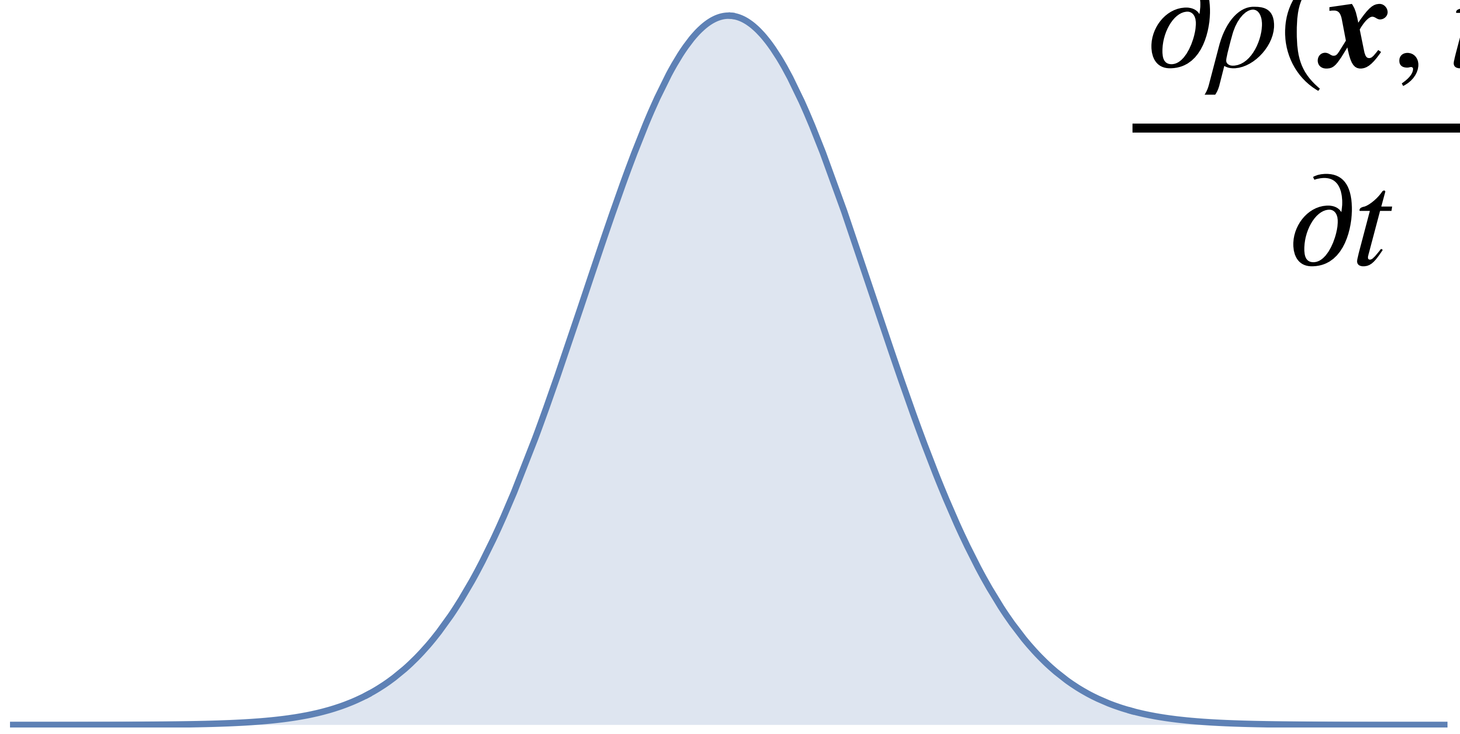


$$\frac{d \ln \rho(\boldsymbol{x}, t)}{dt} = - \nabla \cdot \boldsymbol{v}$$

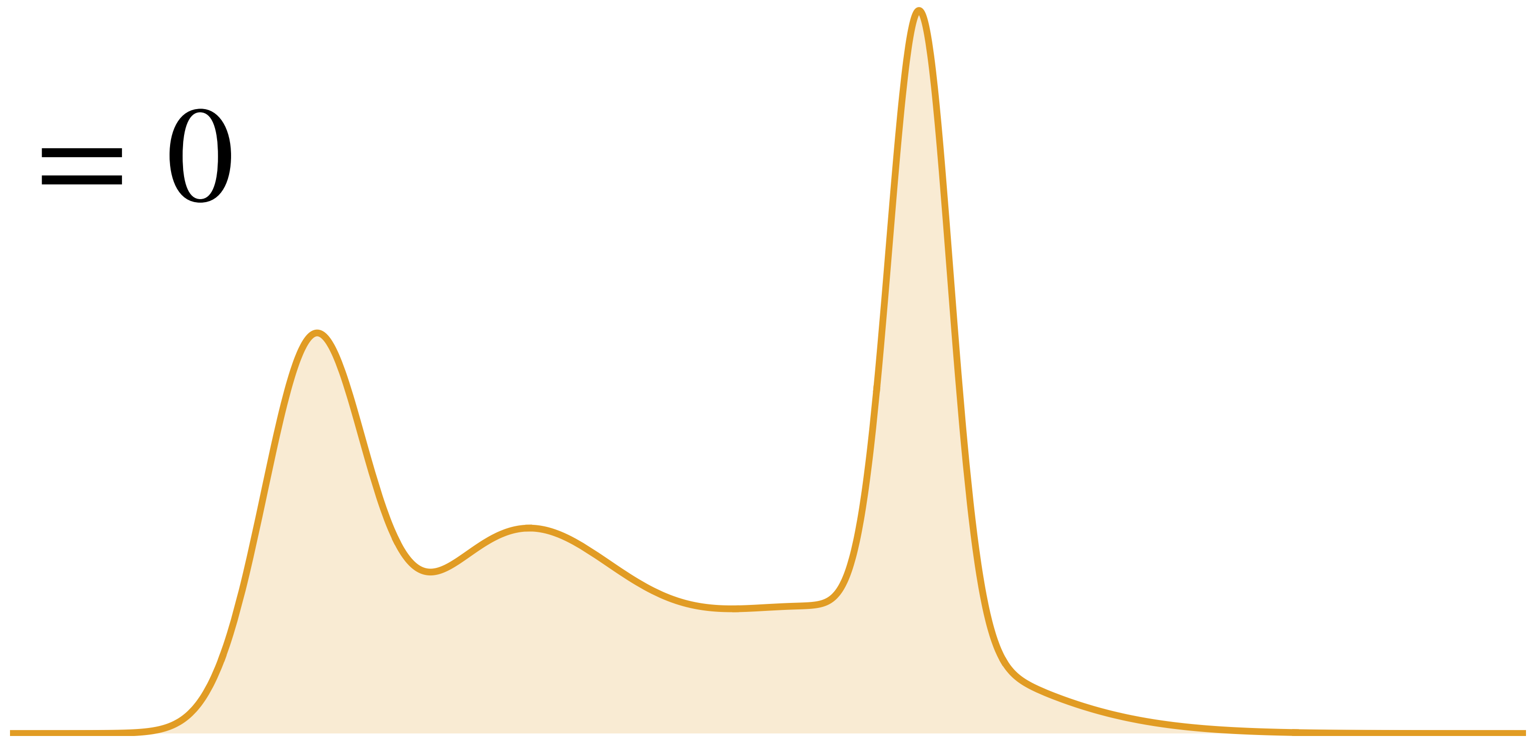
$$\frac{d}{dt} = \frac{\partial}{\partial t} + \boldsymbol{v} \cdot \nabla$$

"material derivative"

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



Simple density

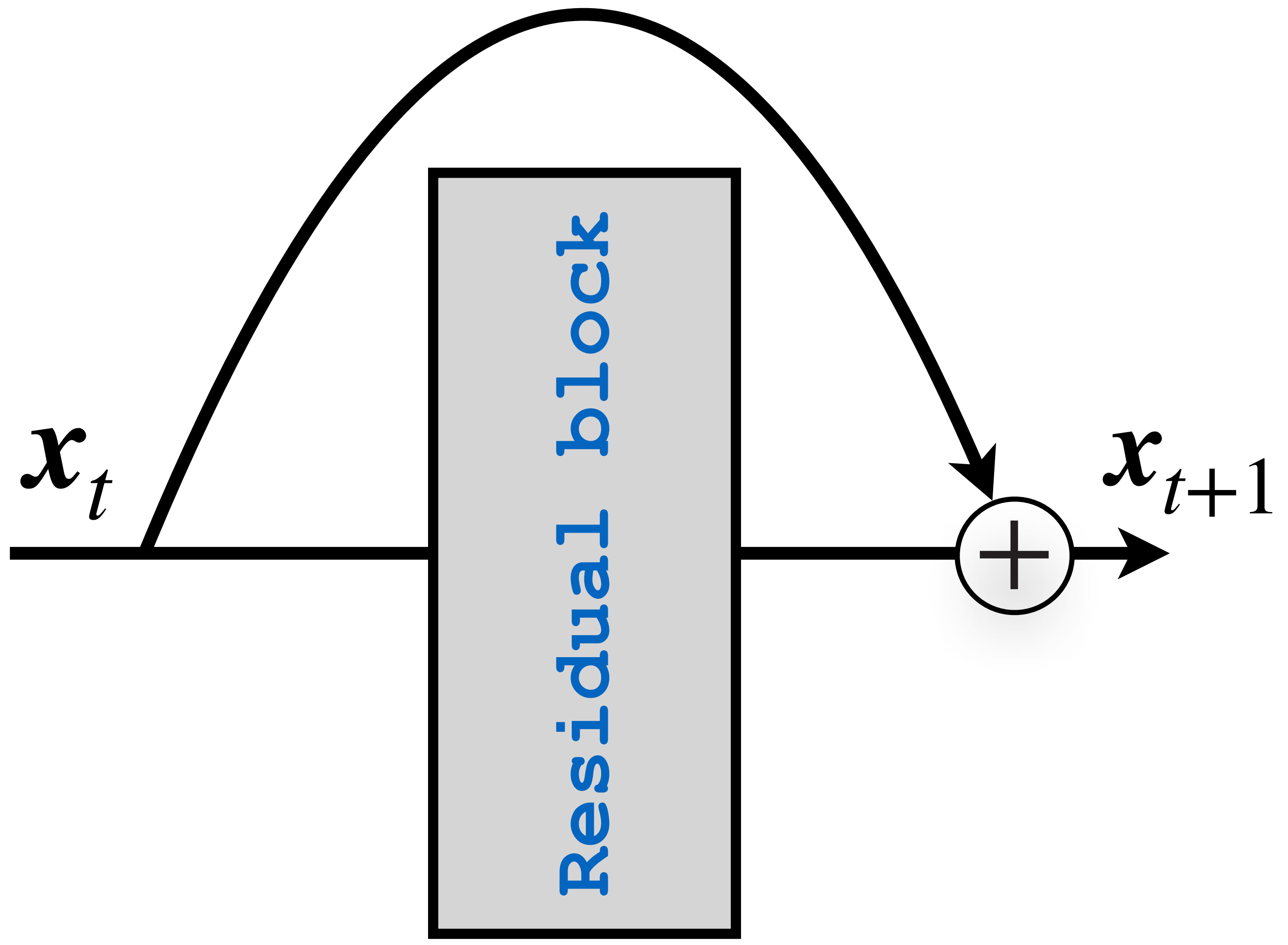


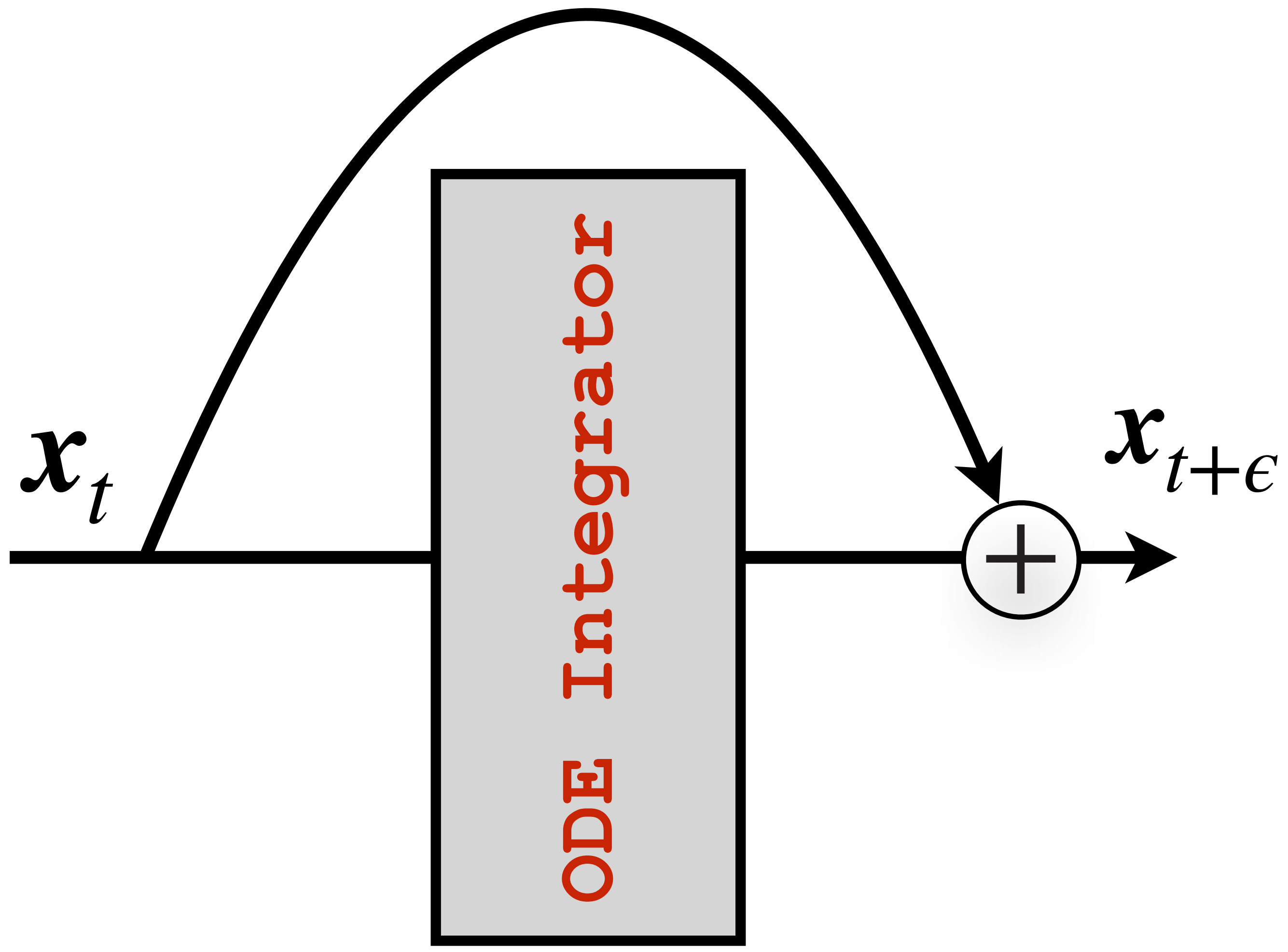
Complex density

Neural Ordinary Differential Equations

Residual network

ODE integration





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

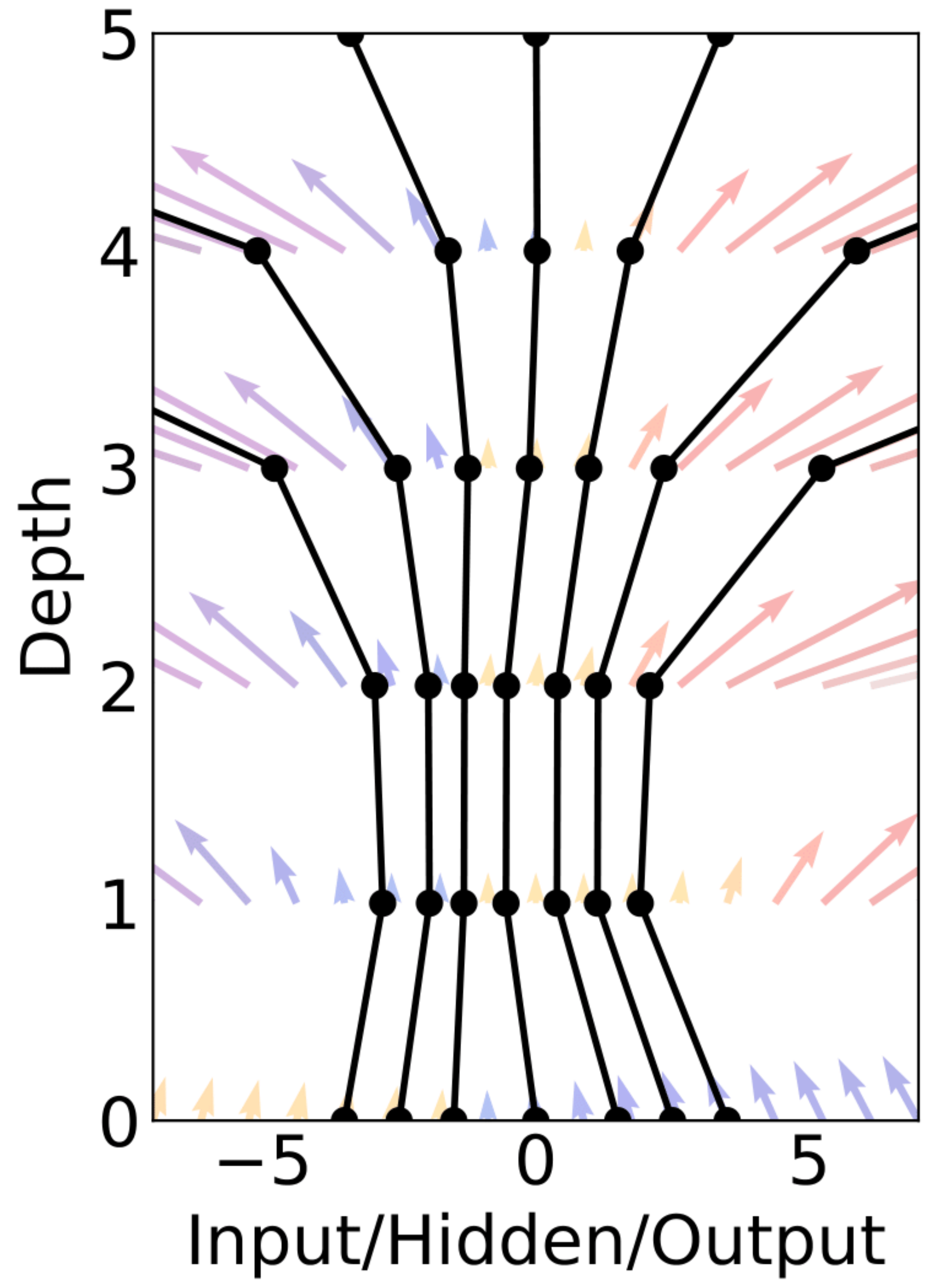
$$dx/dt = v(x)$$

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

Chen et al, 1806.07366

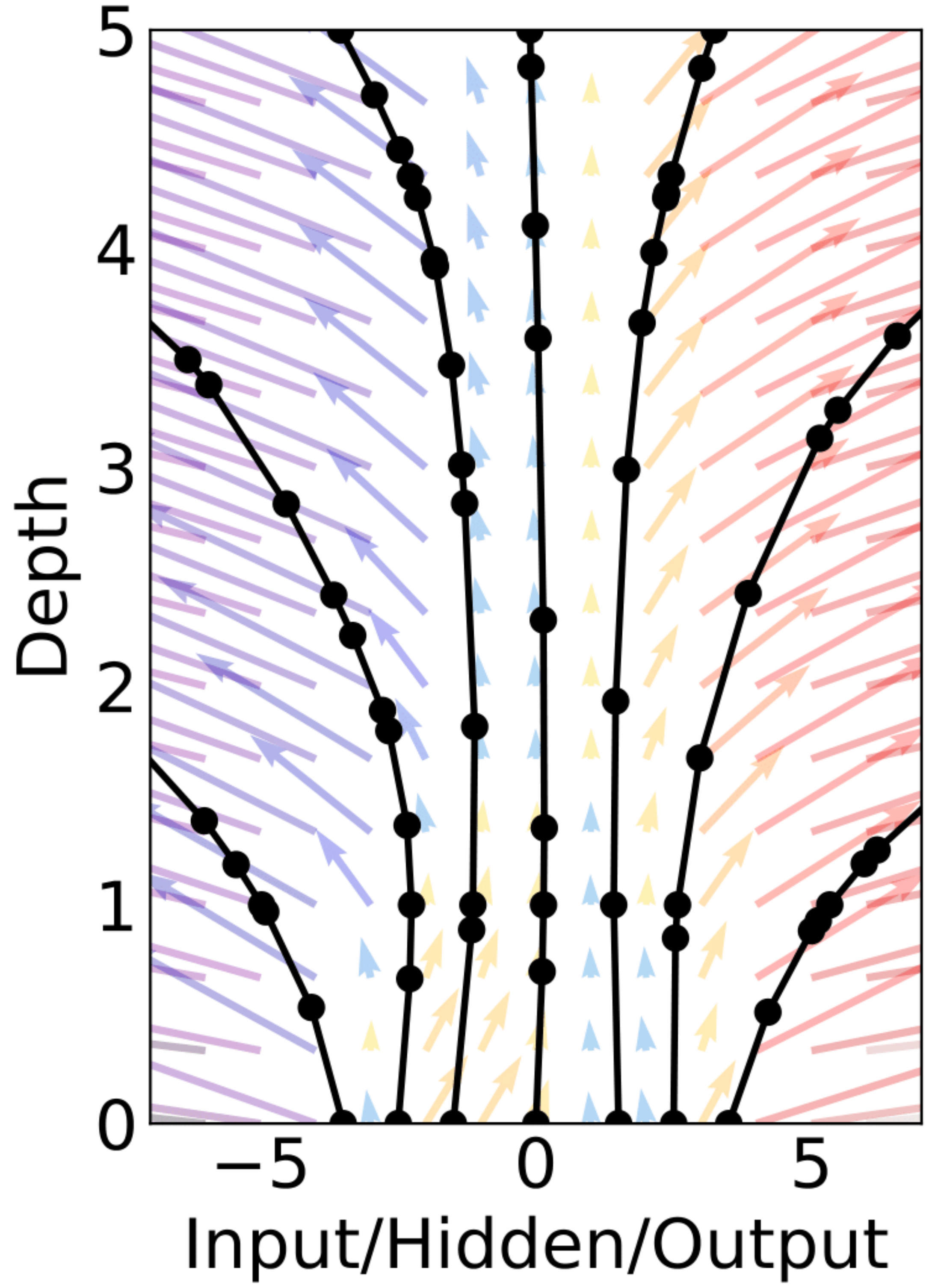
Neural Ordinary Differential Equations

Residual network



$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}(\boldsymbol{x}_t)$$

ODE integration



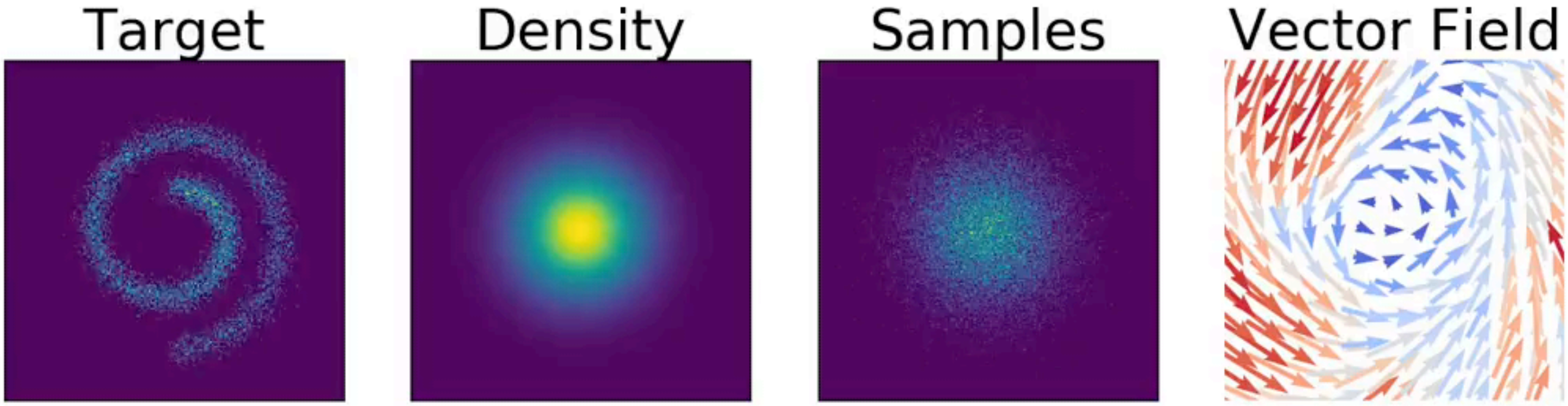
$$d\boldsymbol{x}/dt = \boldsymbol{v}(\boldsymbol{x})$$

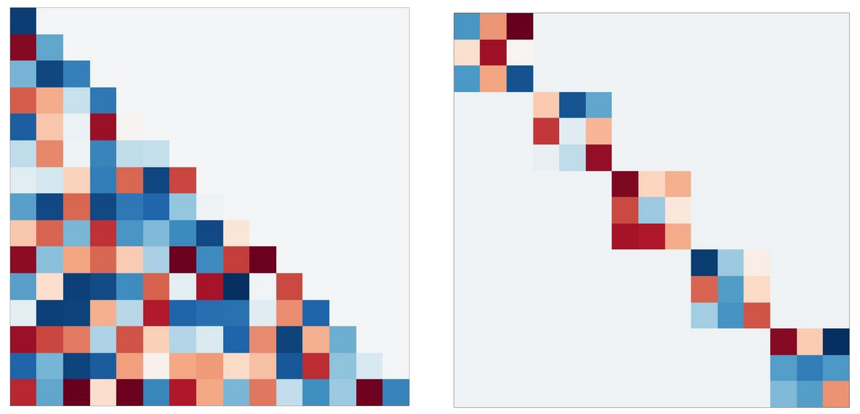
Harbor el al 1705.03341
Lu et al 1710.10121,
E Commun. Math. Stat 17'...

Chen et al, 1806.07366

Continuous normalizing flows implemented with NeuralODE

Chen et al, 1806.07366, Grathwohl et al 1810.01367

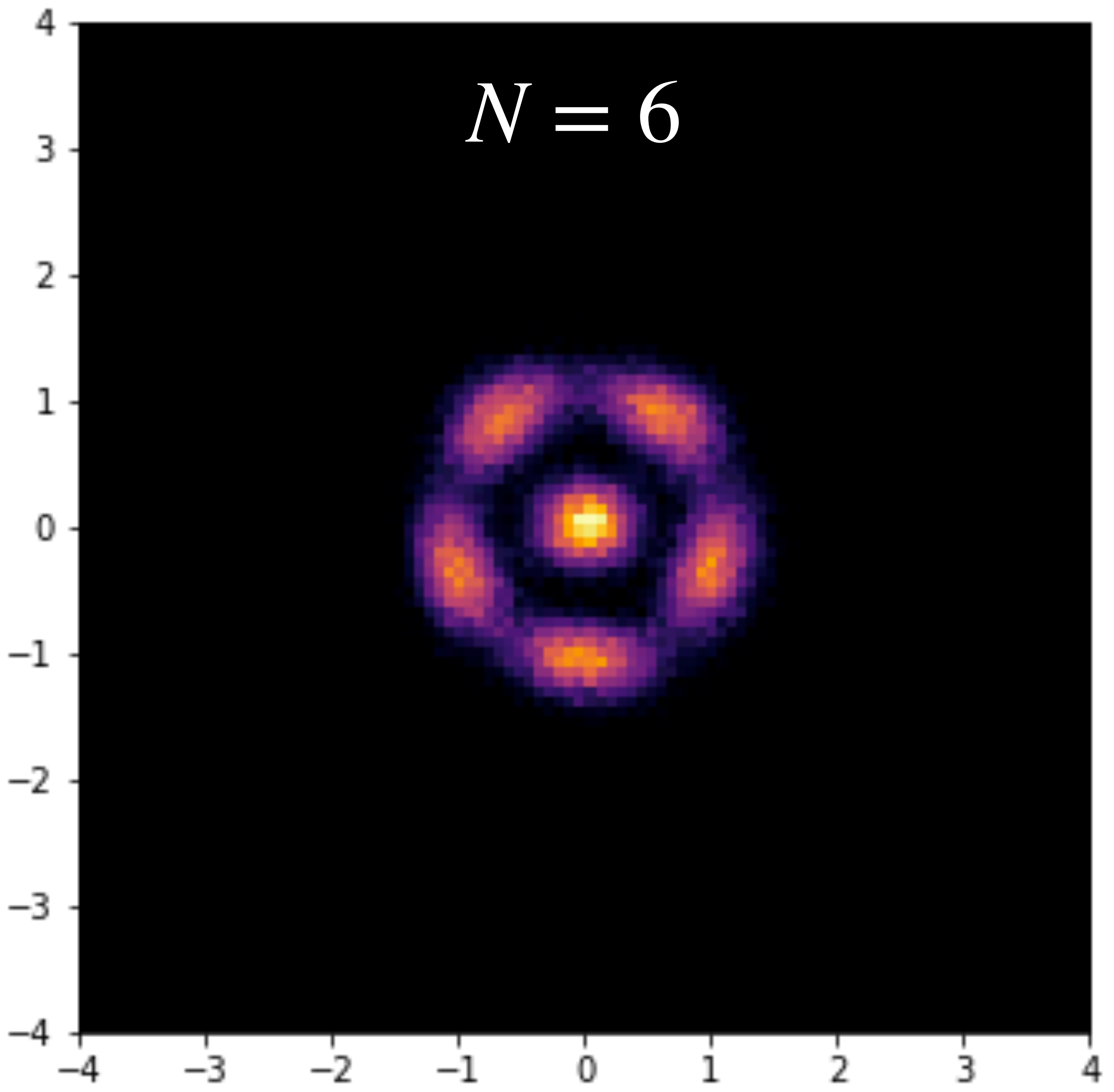


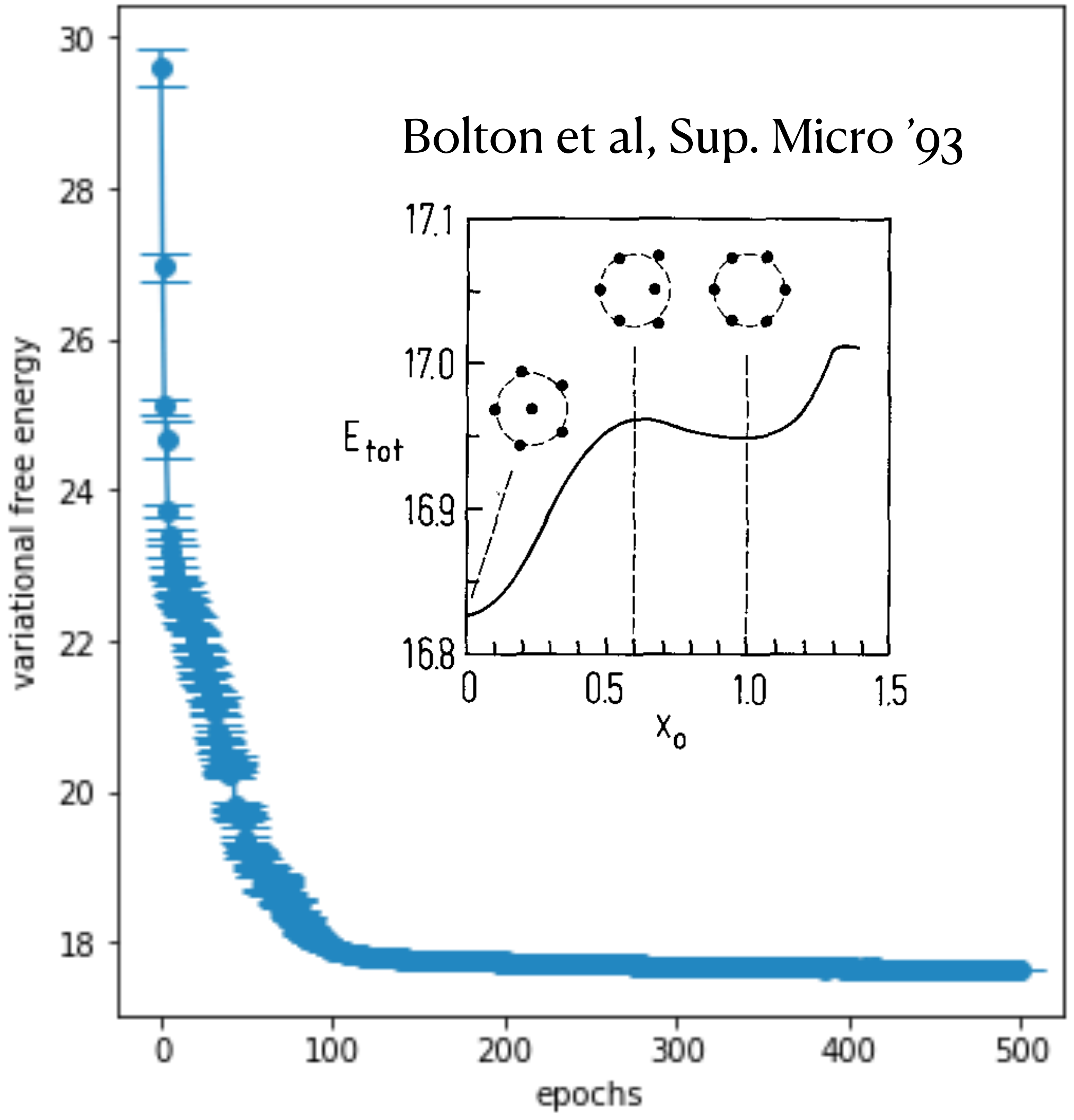


Continuous normalizing flow have no structural constraints on the transformation Jacobian

Tutorial: Classical Coulomb gas in a harmonic trap

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





https://github.com/fermiflow/FermiFlow/blob/github/classical_coulomb_gas.ipynb

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.

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

A few words about tooling

HIPS/autograd

theano

Zygote

MindSpore

PyTorch

TensorFlow

SciML

Keras

NiLang

Taichi

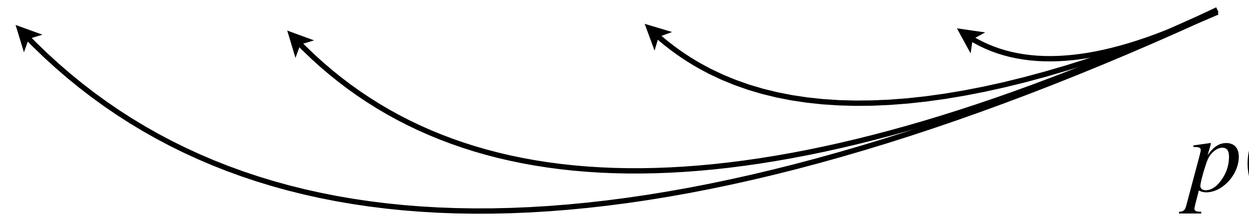
Differentiable programming frameworks

Autoregressive model

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

Language: Casual transformer 1706.03762

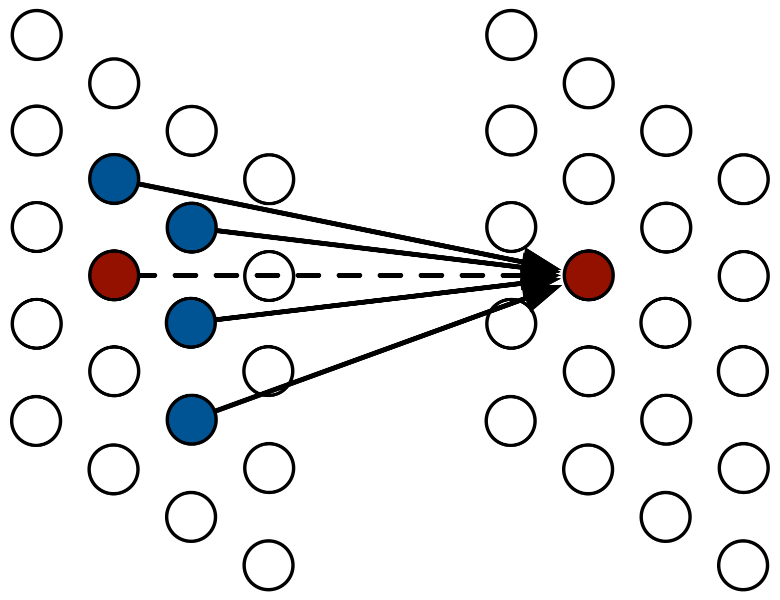
"... quick brown fox jumps ..."



$p(jumps \mid \ldots)$

Speech: WaveNet 1609.03499

Image: PixelCNN 1601.06759



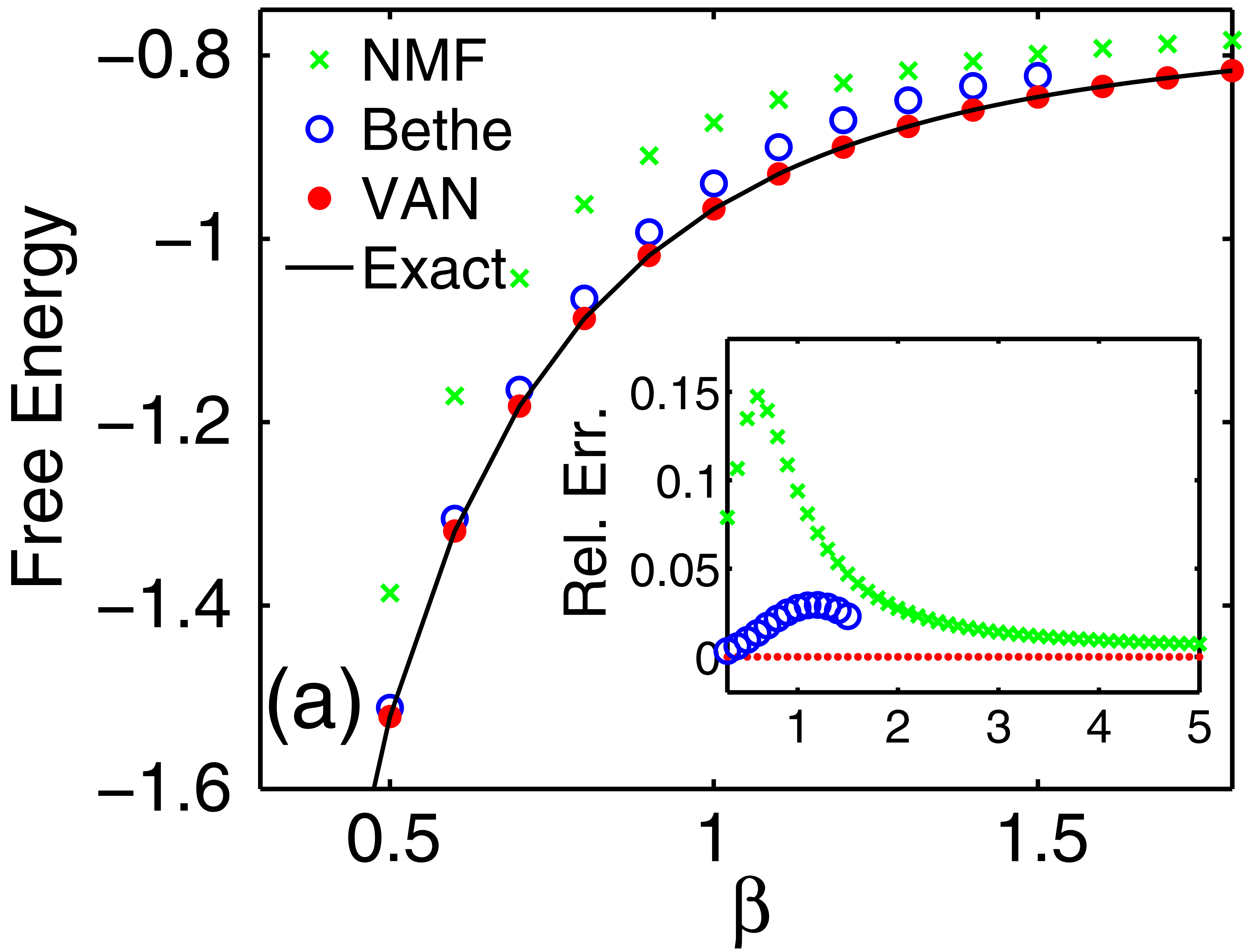
Exercise

Wait, Isn't WaveNet a normalizing flow?

Hint: read Papamakarios et al,1705.07057
and van den Oord et al,1711.10433

Variational autoregressive networks

Sherrington-Kirkpatrick spin glass



Conventional approaches

Naive mean-field factorized probability $p(\boldsymbol{x}) = \prod_i p(x_i)$

Bethe approximation pairwise interaction $p(\boldsymbol{x}) = \prod_i p(x_i) \prod_{(i,j)\in E} \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$

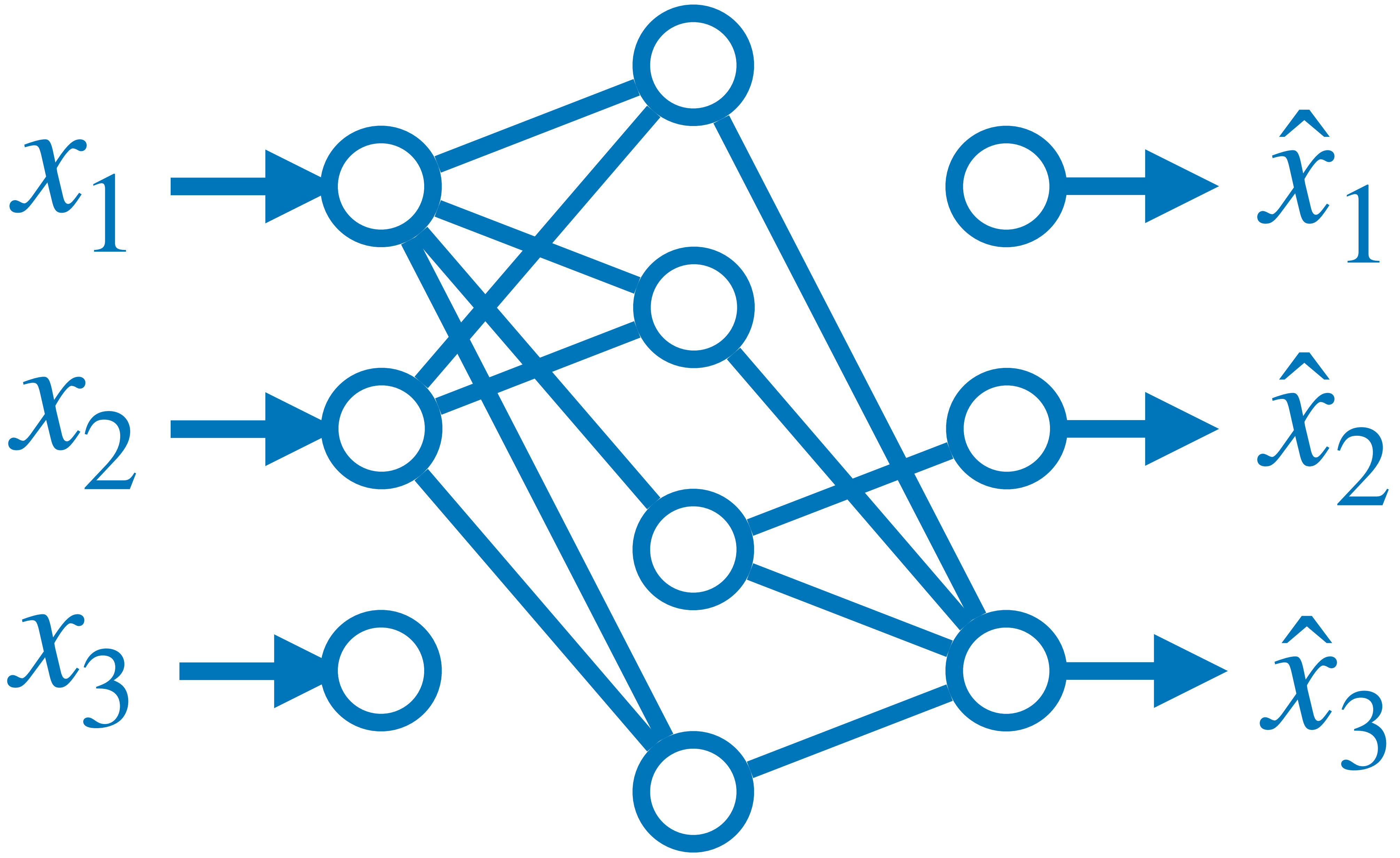
Variational autoregressive network

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

Wu, LW, Zhang, PRL '19

github.com/wdphy16/stat-mech-van

Implementation: autoregressive masks



Masked Autoencoder
Germain et al, 1502.03509

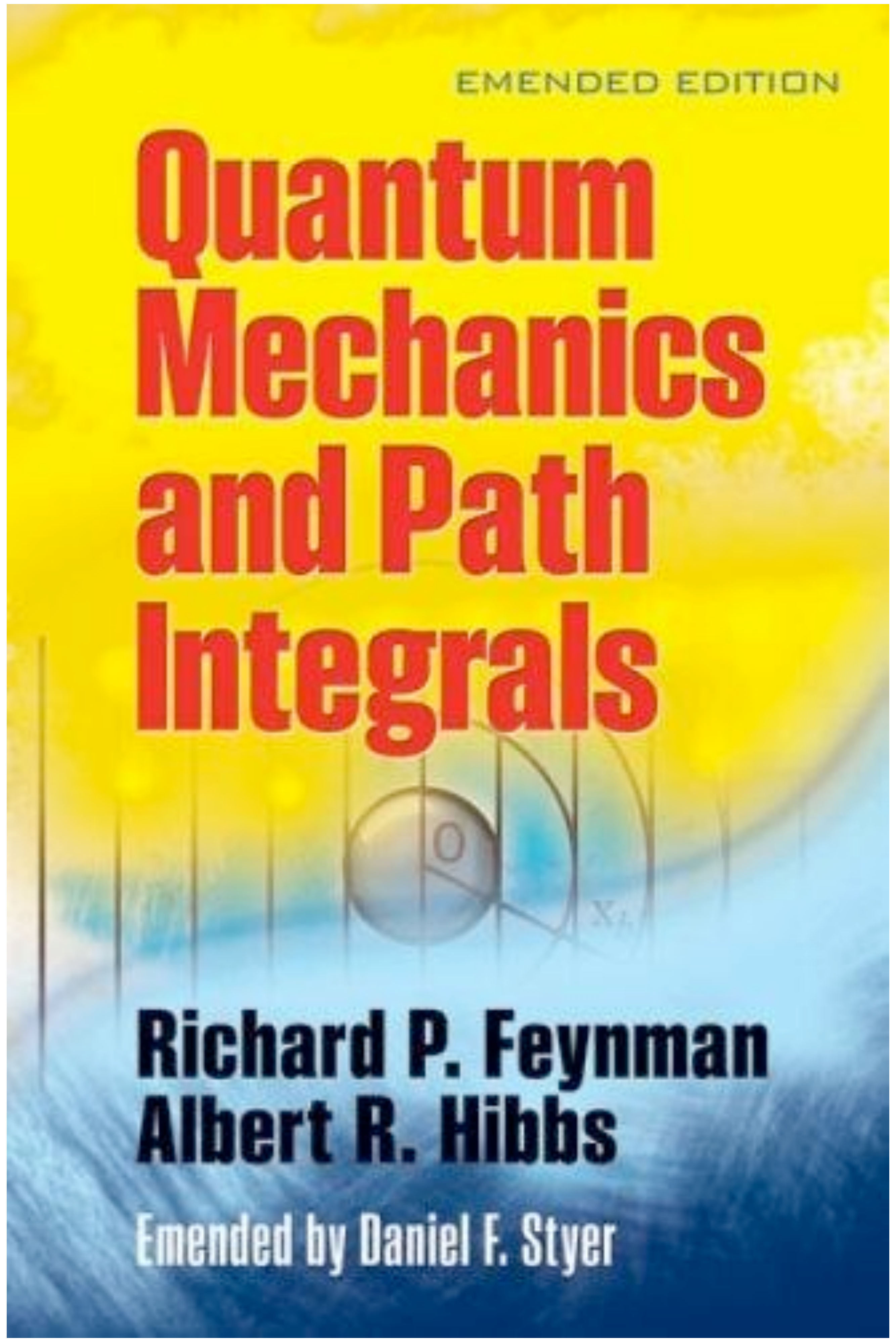
$p(x_1) = \text{Bern}(\hat{x}_1)$ $\quad$ $p(x_2 \mid x_1) = \text{Bern}(\hat{x}_2)$ $\quad$ $p(x_3 \mid x_1, x_2) = \text{Bern}(\hat{x}_3)$

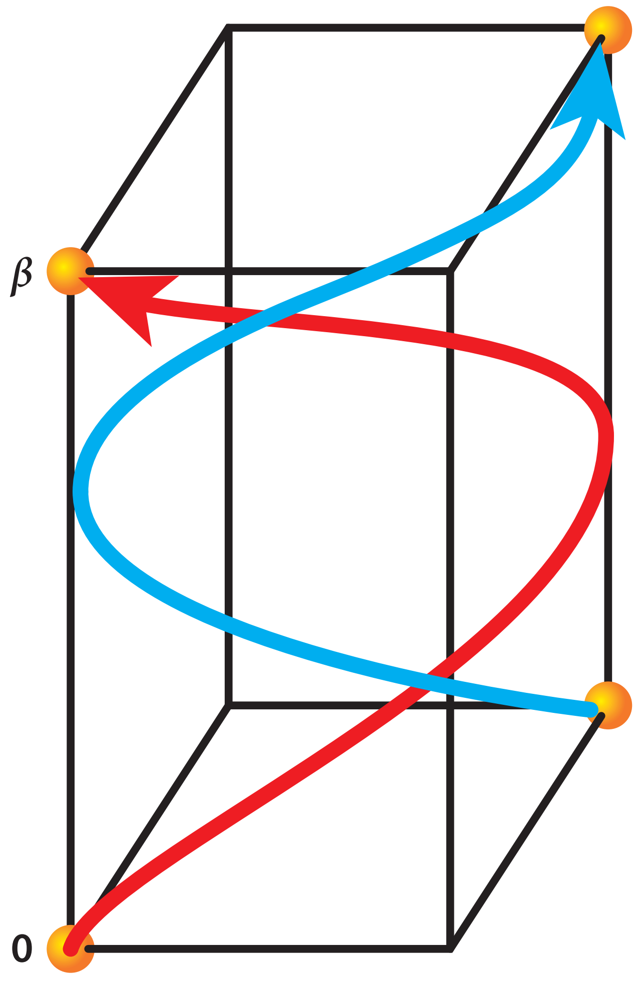
Other examples: PixelCNN, van den Oord et al, 1601.06759 Casual transformer, 1706.03762

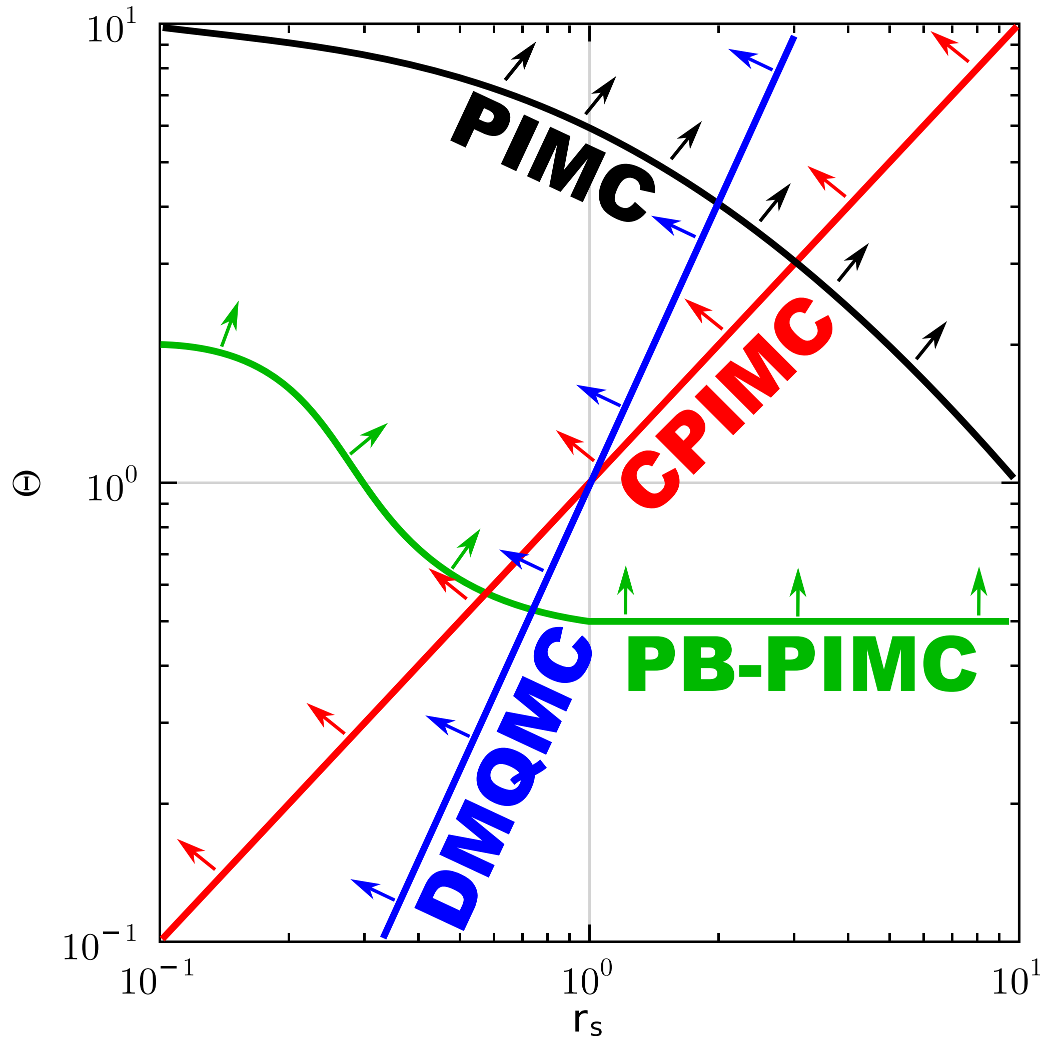
Other ways to implement autoregressive models: recurrent networks

How about quantum systems ?

Quantum-to-classical mapping







d+1-dim
spacetime integral

$$Z = \mathrm{Tr}(e^{-\beta H}) = \int d^{d+1}\boldsymbol{x}\ \cdots$$

However, the “weight”
may not be positive definite.
Sign problem!

The quantum variational free-energy approach

Gibbs–Bogolyubov-Feynman-Delbrück–Molière variational principle

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

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

Exercise

Prove $F[\rho] \geq -k_B T \ln Z$

where $Z = \mathrm{Tr}(e^{-H/k_B T})$

Search "Quantum relative entropy" on wikipedia

Exercise

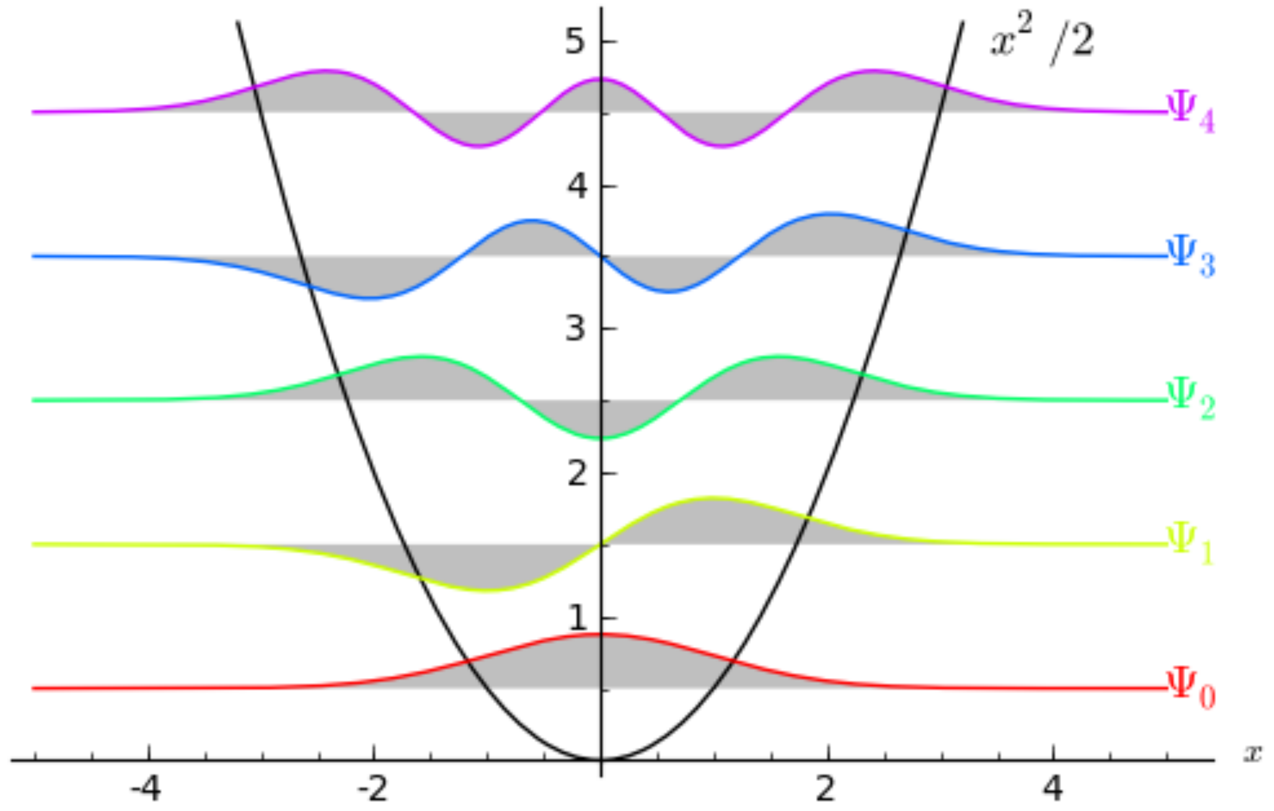
Think about how to solve the quantum Coulomb gas problem using this principle.

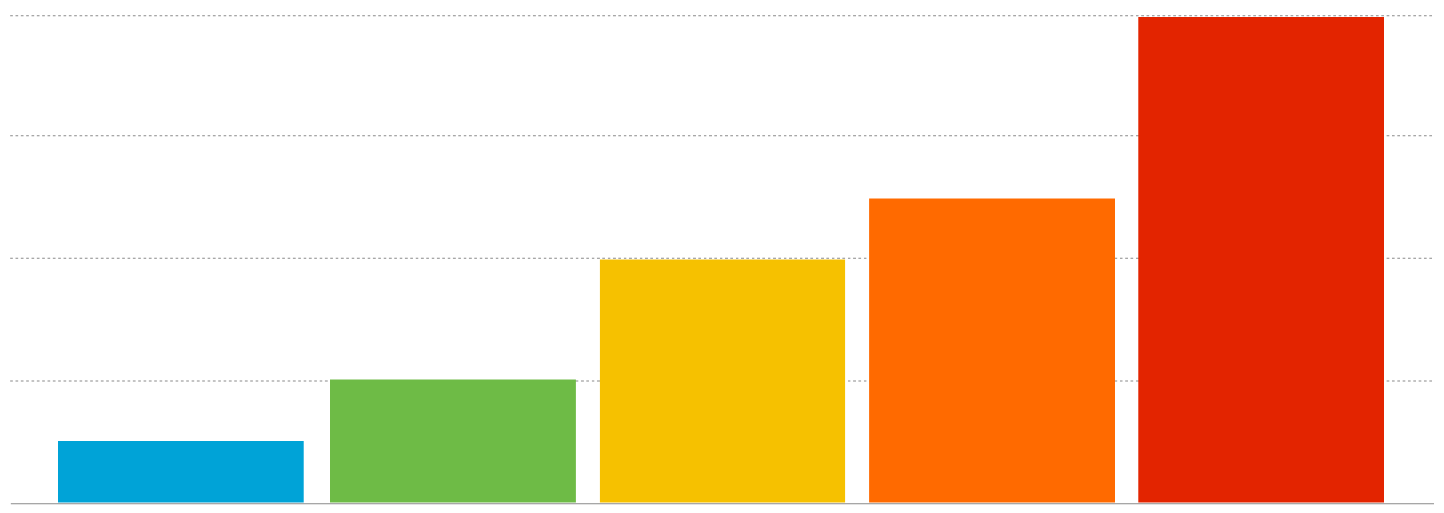
Density matrix

$$\rho = \sum_n \mu_n |\Psi_n\rangle\langle\Psi_n|$$

Quantum states $\Psi_n(\boldsymbol{x}) = \langle \boldsymbol{x} | \Psi_n \rangle$

Classical probability $0 < \mu_n < 1$





$$\langle \Psi_m | \Psi_n \rangle = \delta_{mn}$$

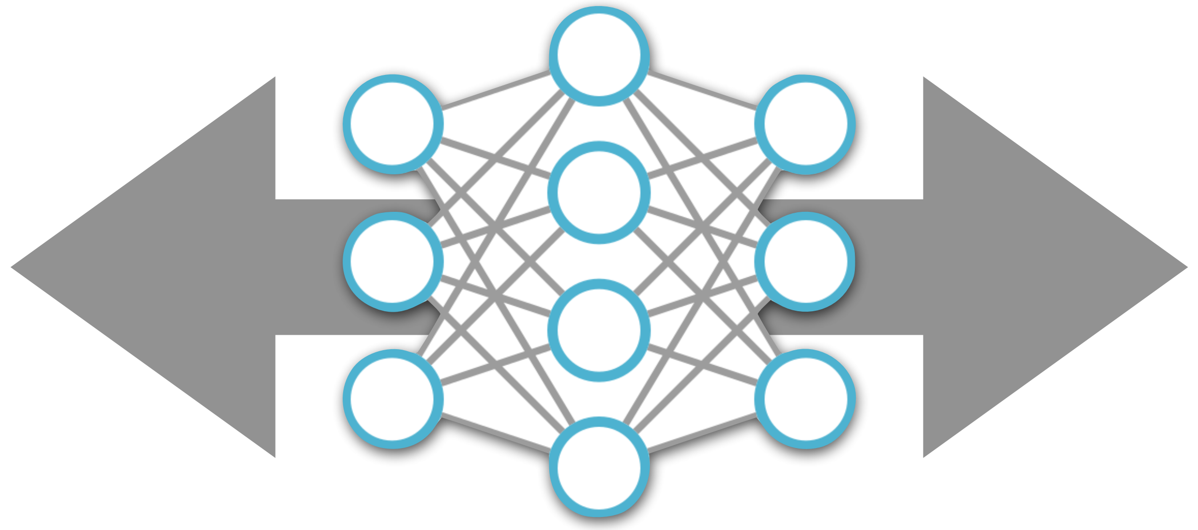
$$\sum_n \mu_n = 1$$

How to represent them ??

Use two deep generative models !!

"Square root" of a normalizing flow

Particle coordinates $\boldsymbol{x}$



$\boldsymbol{z}$ Quasi-particle coordinates

$$\Psi_n(\boldsymbol{x}) = \Phi_n(\boldsymbol{z}) \cdot \left| \det\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} \right) \right|^{\frac{1}{2}}$$

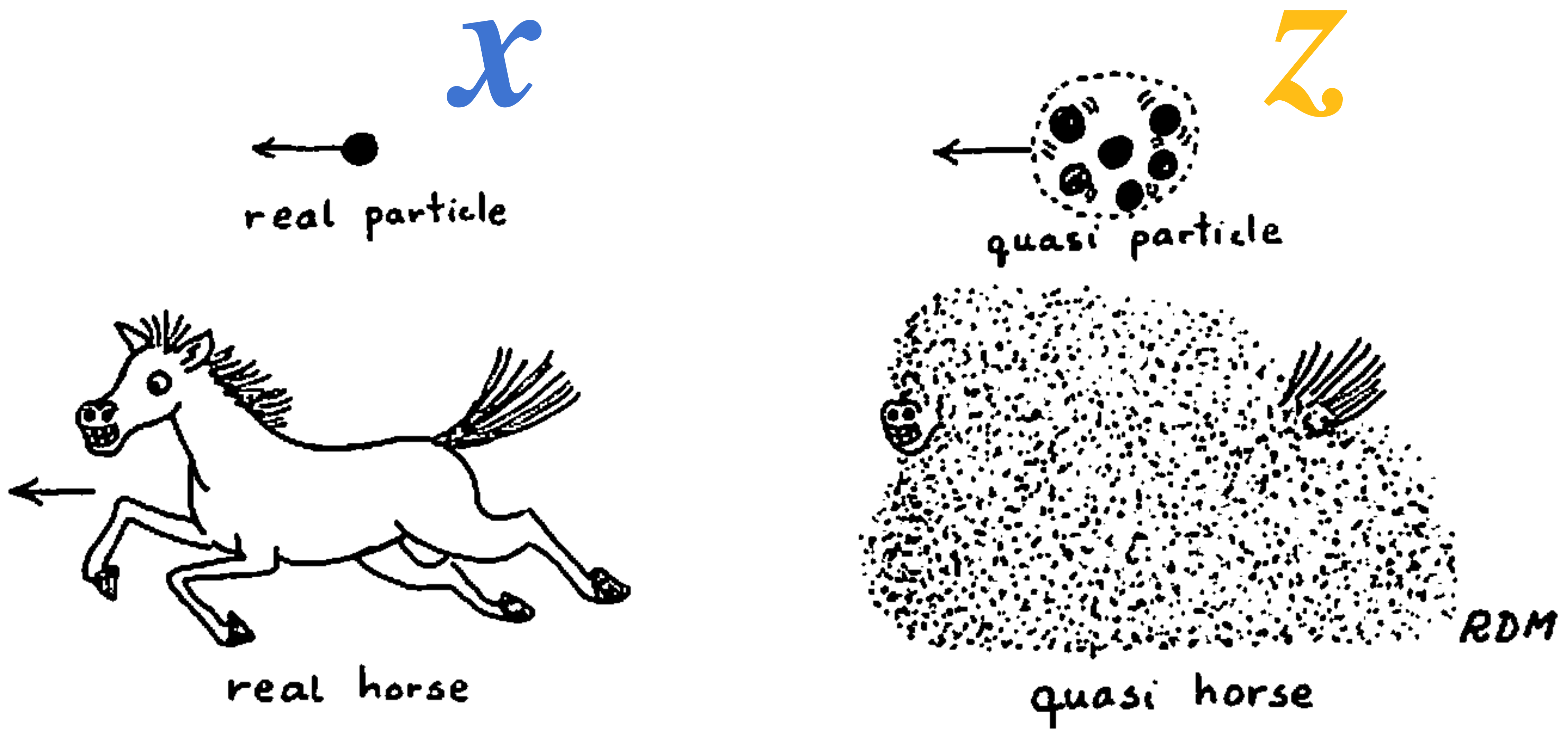
Base states

Jacobian of the flow

The flow implements a *learnable* many-body unitary transformation

hence the name "neural canonical transformation" a classical generalization of Li, Dong, Zhang, LW, PRX '20

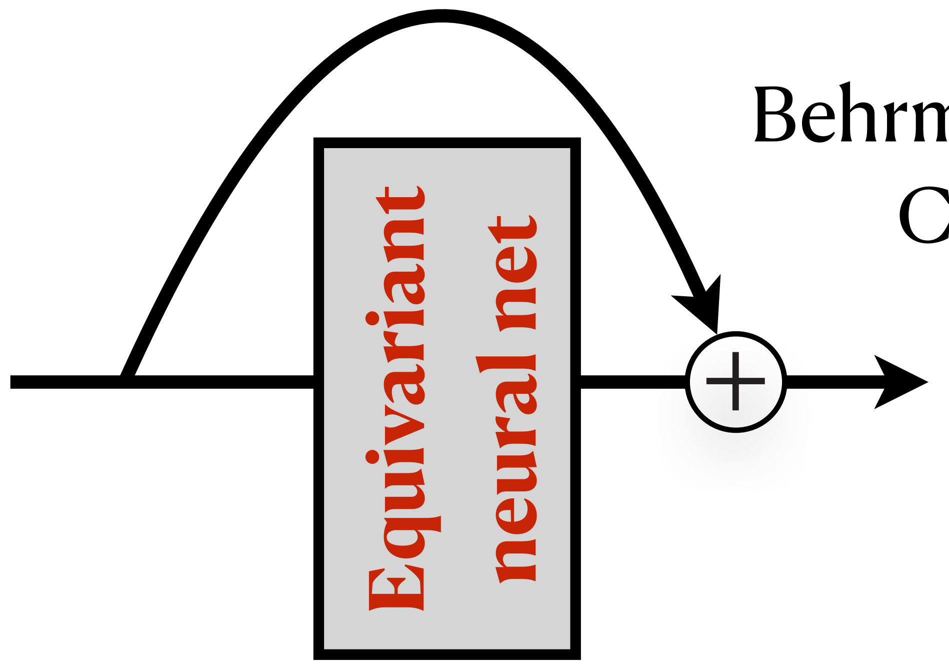
Feynman's backflow in the deep learning era



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

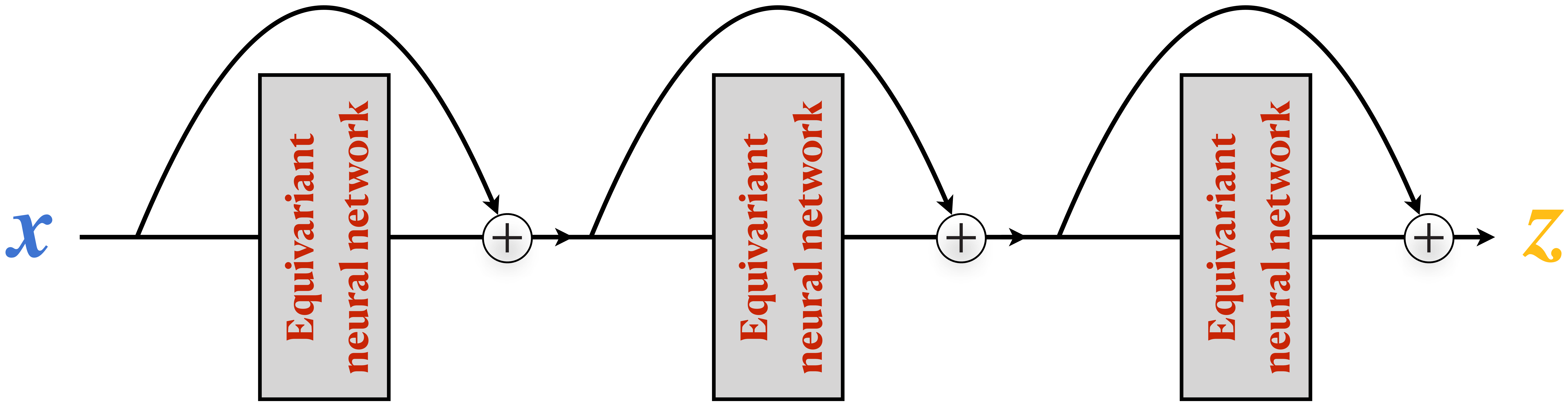
Feynman & Cohen 1956
wavefunction for liquid Helium

1. Backflow can be made unitary (if we track its Jacobian)
2. Backflow is an equivariant residual flow



Behrmann et al, 1811.00995
Chen et al, 1906.02735

Feynman’s backflow in the deep learning era



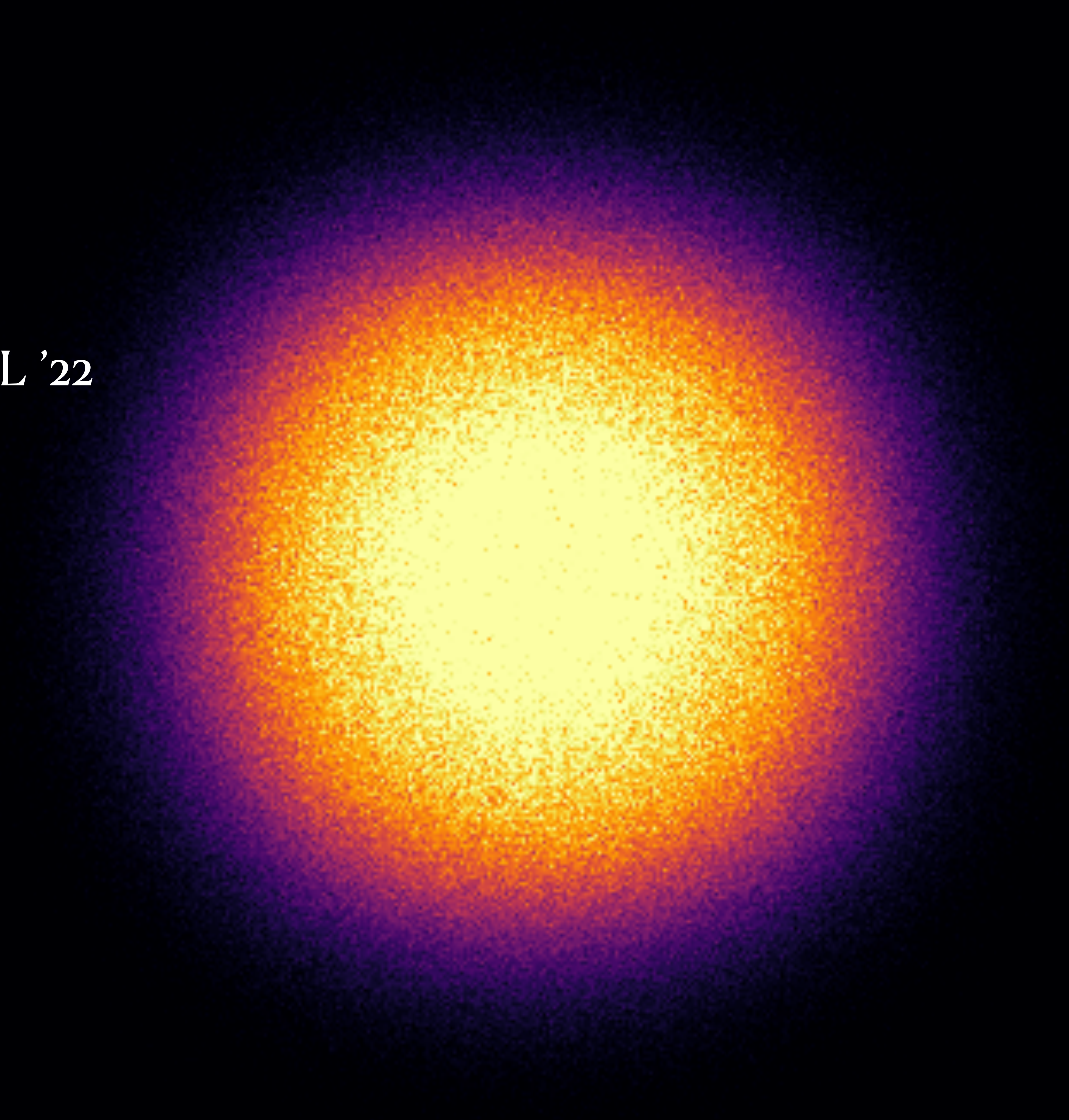
Deep residual networks can be regarded as discretization of a continuous dynamics

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

Exercise

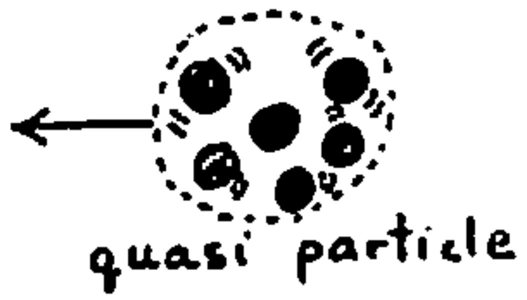
Recall that $\rho = \sum_n \mu_n | \Psi_n \rangle \langle \Psi_n |$, prove

$$\mathrm{Tr}(\rho \ln \rho) = \mathop{\mathbb{E}}_{n \sim \mu_n} \left[\ln \mu_n \right]$$

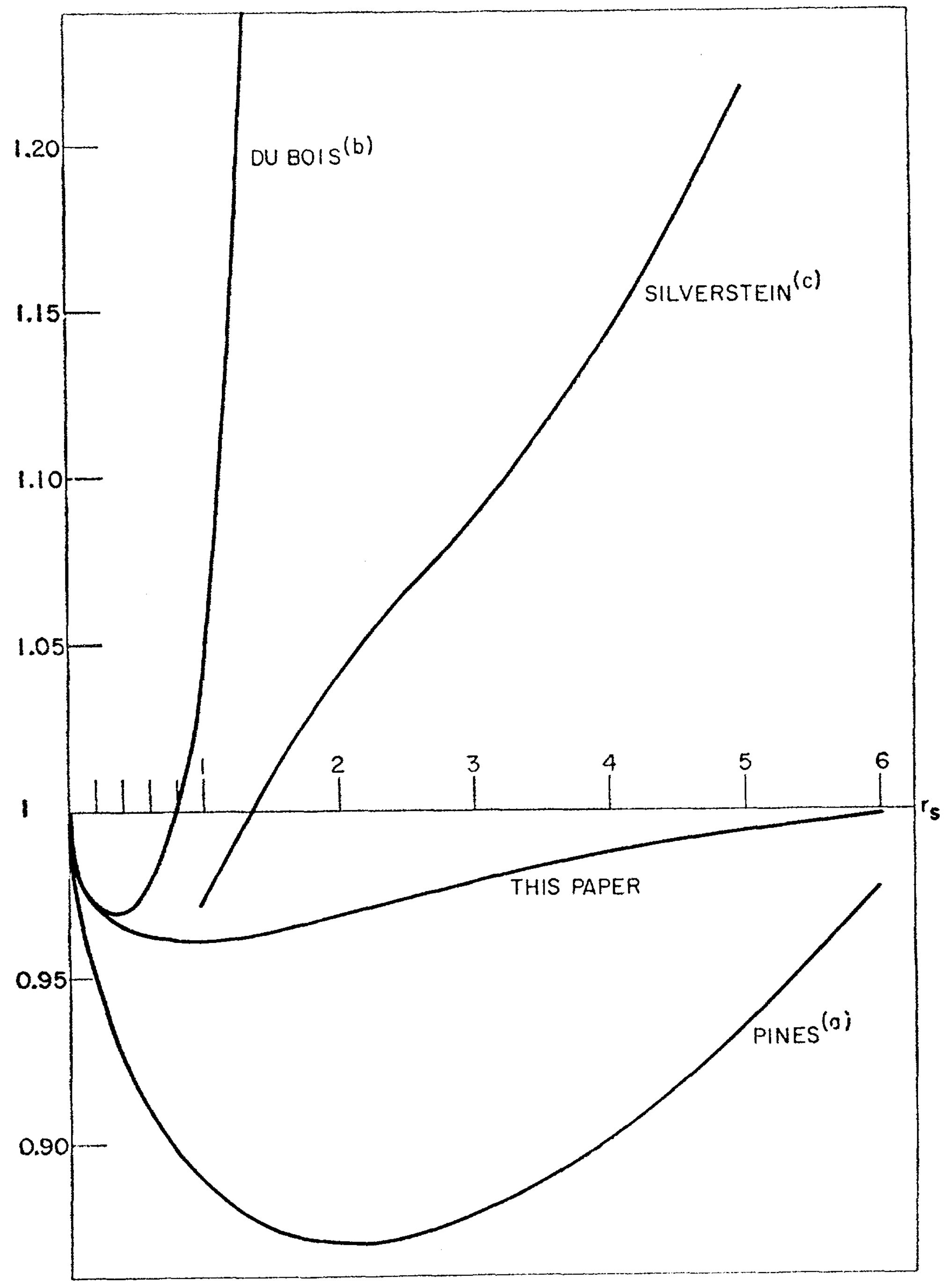
Applications

Uniform electron gas

Hedin Phys. Rev. 1965



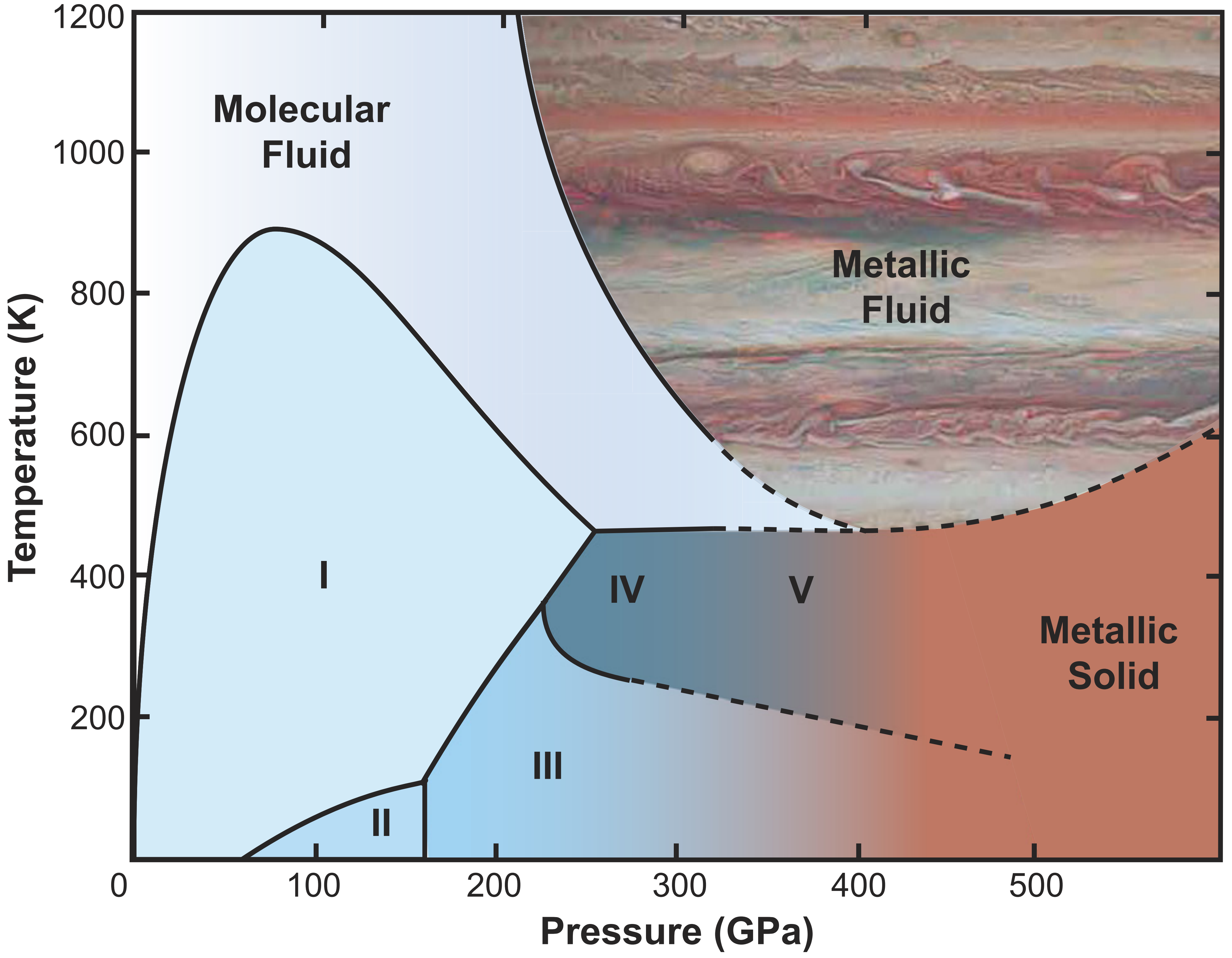
quasi-particle effective mas



Xie, Zhang, LW, arXiv '22

Dense hydrogen

Radiat et al, Extremes, 2020

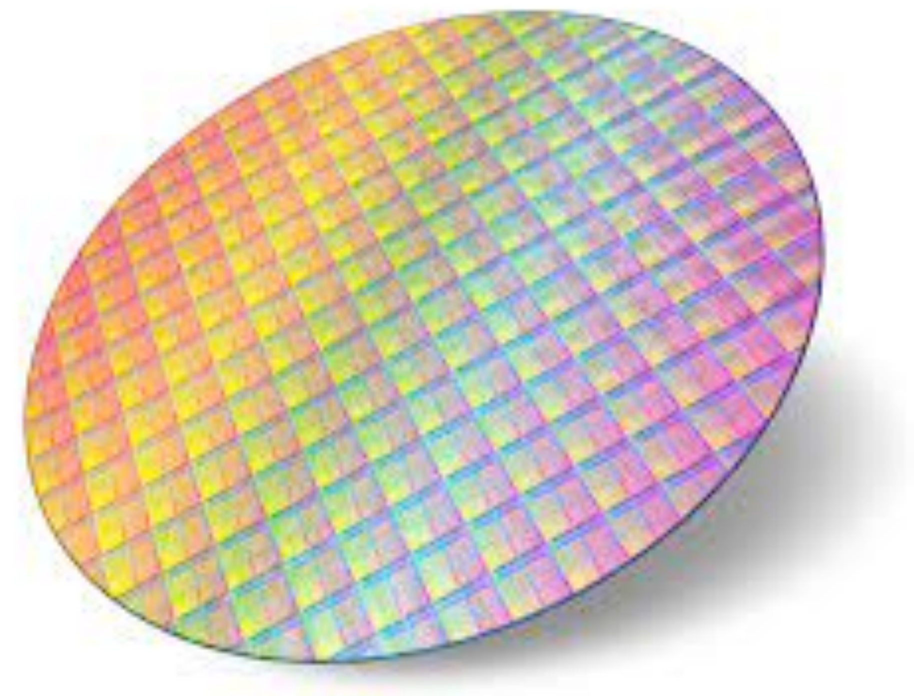


1 to 1000: model architecture based on physics, pretraining, large scale optimization...

Triumph of condensed matter physics

Insulators

Metal

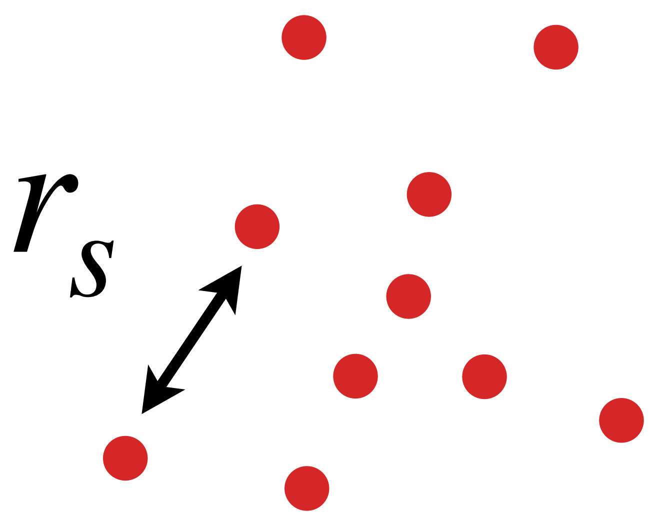


Semiconductors

Why metal is metal ?

Uniform electron gas

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



r_s of typical metals, Richard Martin, *Electronic structure*

$Z=1$	$Z=2$	$Z=1$	$Z=2$	$Z=3$	$Z=4$
Li 3.23	Be 1.88			B	C 1.31
Na 3.93	Mg 2.65			Al 2.07	Si 2.00
K 4.86	Ca 3.27	Cu 2.67	Zn 2.31	Ga 2.19	Ge 2.08
Rb 5.20	Sr 3.56	Ag 3.02	Cd 2.59	In 2.41	Sn 2.39
Cs 5.63	Ba 3.69	Au 3.01	Hg 2.15	Tl	Pb 2.30

Metal density $2 < r_s < 6$: Coulomb repulsion is nonperturbative compared to kinetic energy

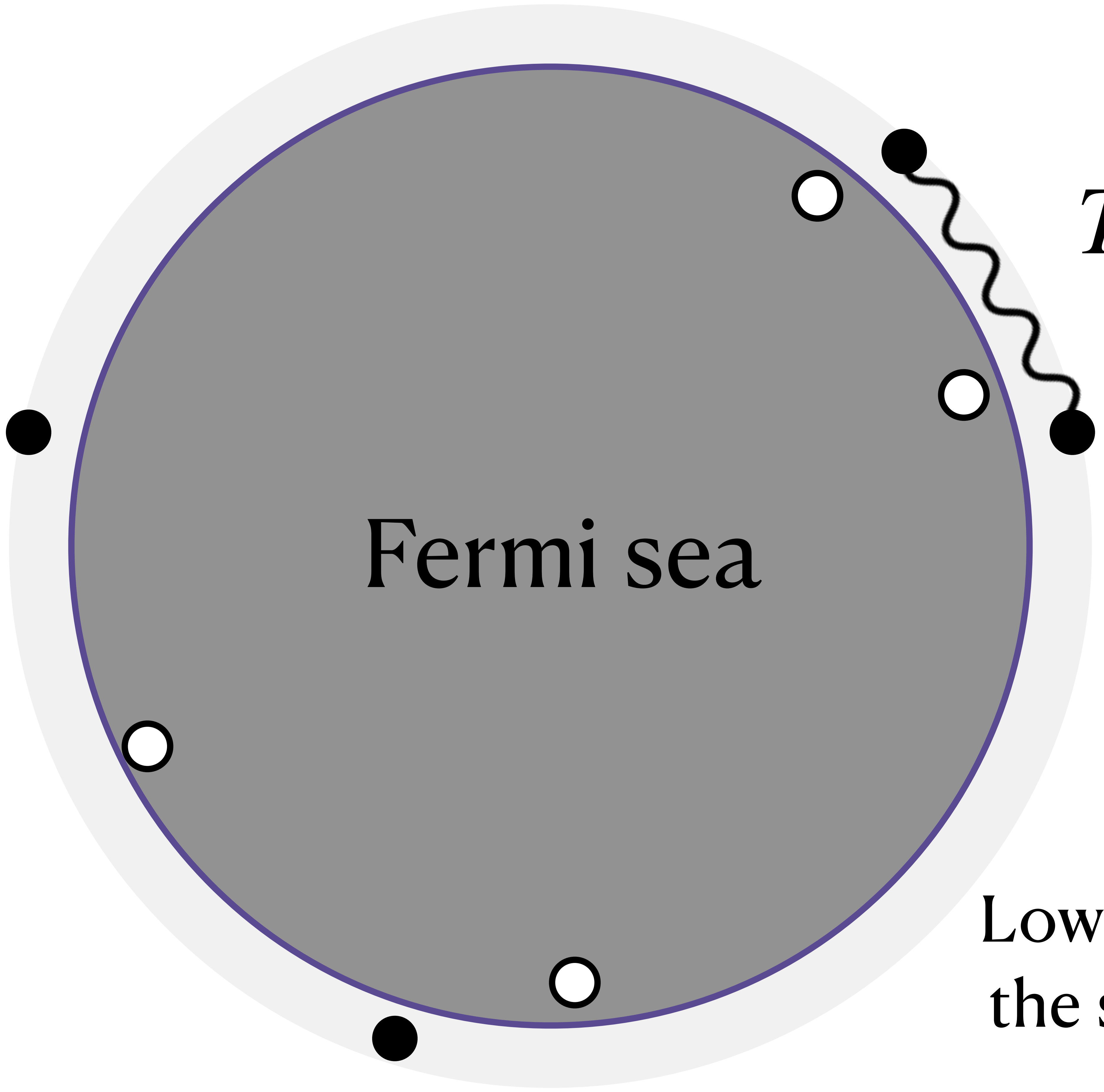
《物理》 2010年第8期

物理学中的演生现象

张广铭[1,†]　于　渌[2,3]

些非正常金属态可以被认为是费米液体的某些不稳定的、对称性破缺的基态. 另一方面，费米液体理论也十分令人费解，因为普通金属中电子之间的库仑相互作用能和费米能是一个量级，同时比费米能量附近的能级间距要大很多. 微扰理论对如此强的相互作用已不再适应，很难相信一个如此强大的相互作用的多电子系统能与一个无相互作用准粒子系统的行为相像. 然而，自然界本身一遍又一遍地提醒我们：尽管有强大的库仑相互作用，金属的低能行为仍与一个自由准粒子系统类似，这又是重正化群思想再奏凯歌！直到 20 世纪 80 年代末，由于分数量子

Landau fermi liquid theory



$$T \ll T_F \lesssim \frac{e^2}{r_s}$$

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

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

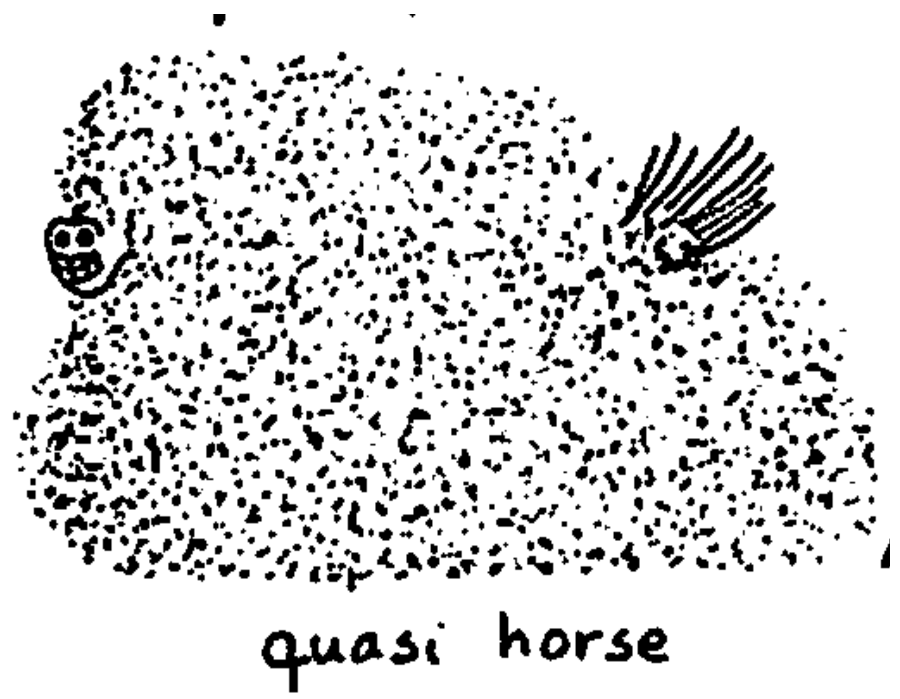
Physics happens around the Fermi surface with strongly constrained phase-space

Have we known everything about a Fermi liquid ?

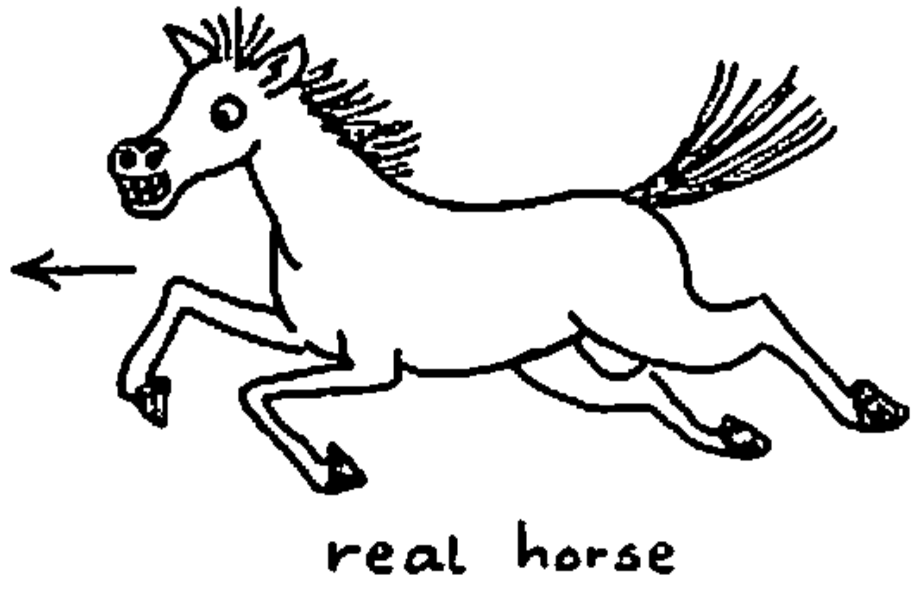
No!

Quasi-particles effective mass

$$\frac{m^*}{m} =$$



/



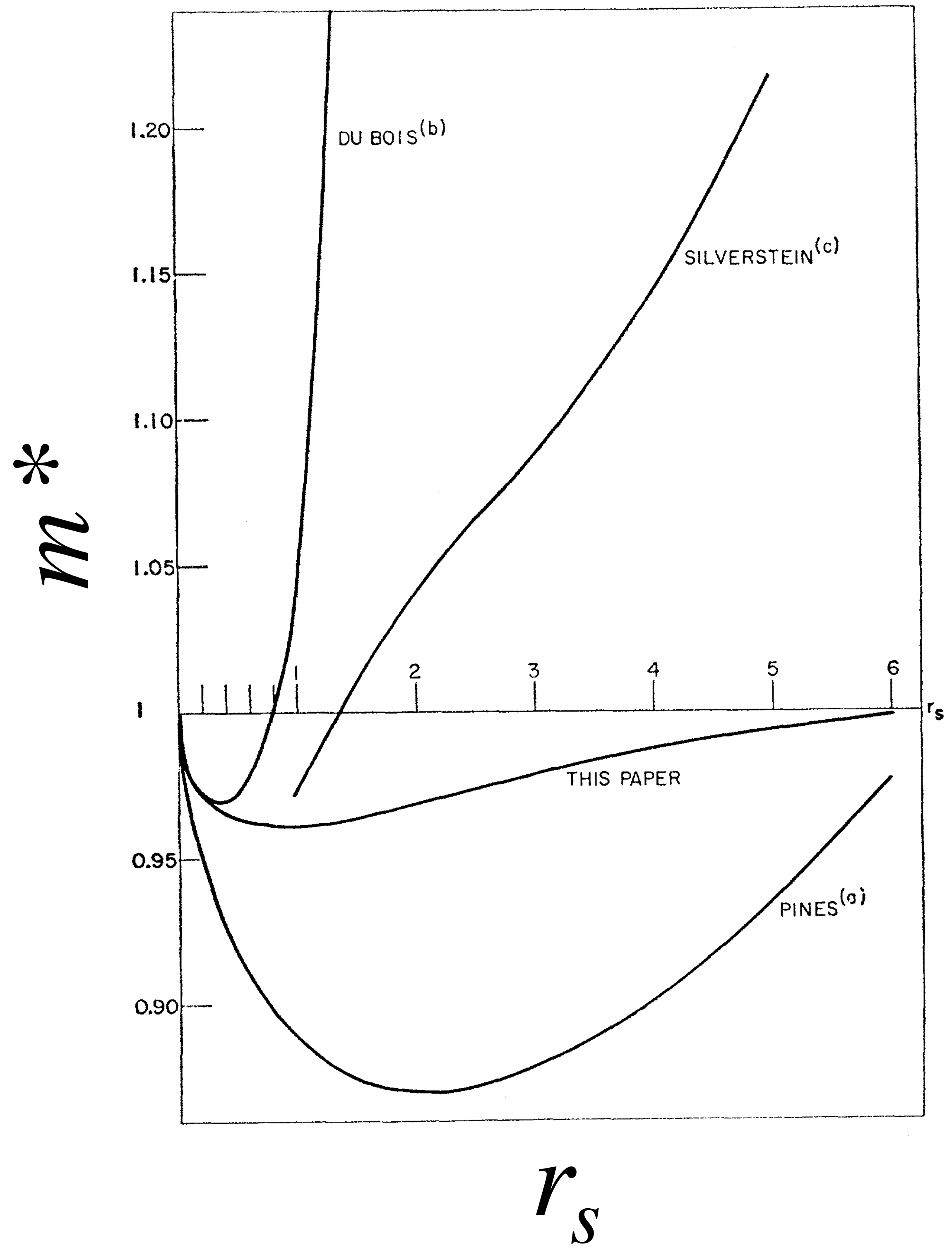
Richard D. Mattuck
A Guide to Feynman Diagrams in the Many-body Problem

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

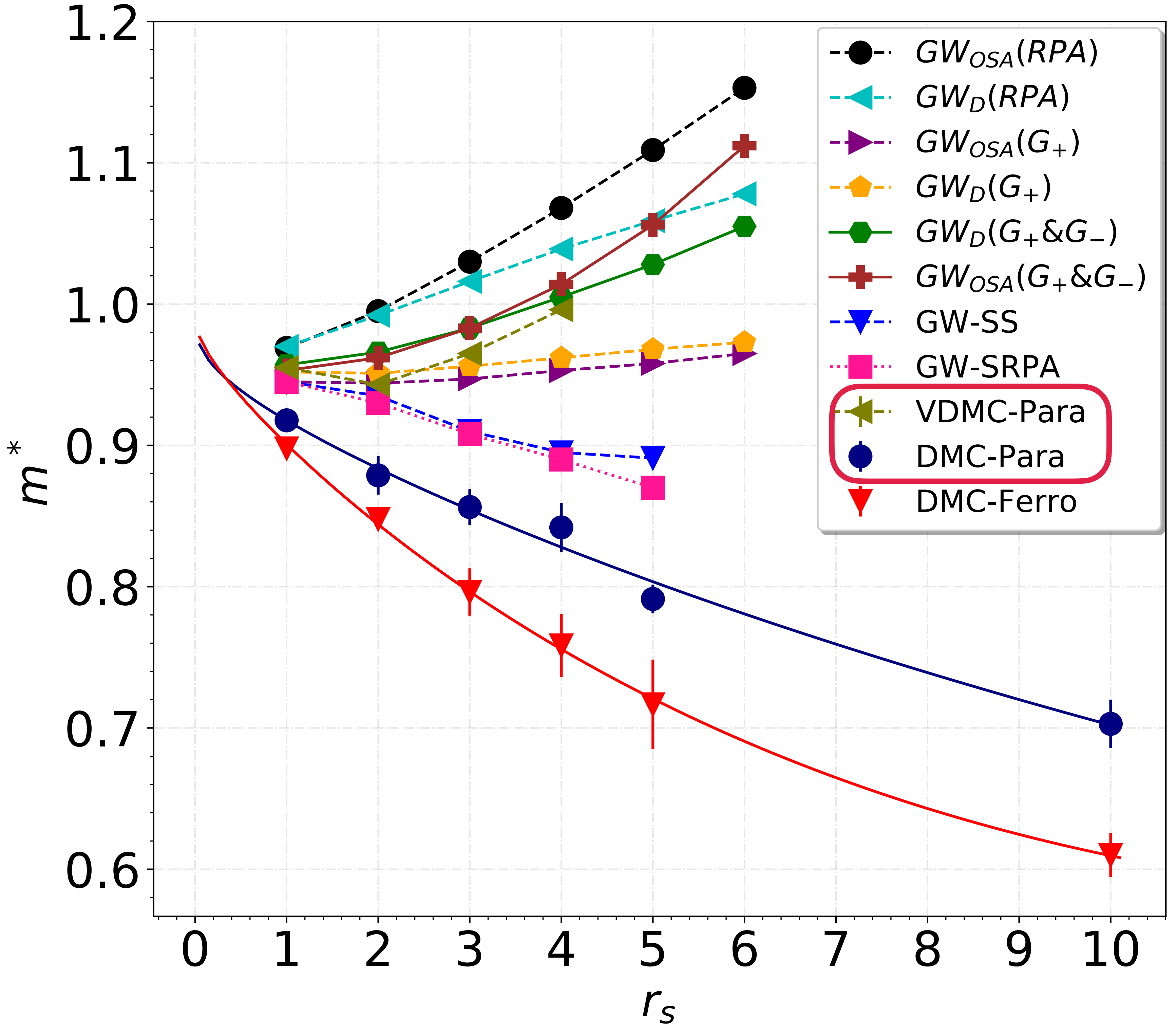
$N(0)$	s	c_V	χ
Density of states	entropy	specific heat	magnetic susceptibility

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...

m^* from low temperature entropy

Eich, Holzmann, Vignale, PRB '17

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

$$\cancel{m^*/m = \left(1-\frac{\partial\Sigma}{\partial\omega}\right)\left(1+\frac{m}{k}\frac{\partial\Sigma}{\partial k}\right)^{-1}}$$

$$\cancel{m^*/m = k_F/(d\varepsilon/dk)_{k_F}}$$

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

interacting electrons

noninteracting electrons

Not an easy task due to the lack of reliable methods
for low-temperature electron gases with intermediate density

computing specific heat also works, but that often requires differentiating (noisy) energies

Deep generative models for the variational density matrix

$$\rho = \sum_{\boldsymbol{K}} \underbrace{p(\boldsymbol{K})}_{\text{Normalized probability distribution}} \underbrace{\left| \Psi_{\boldsymbol{K}} \right\rangle}_{\text{Orthonormal many-electron basis}} \left\langle \Psi_{\boldsymbol{K}} \right|$$

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

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

There will also be interesting twists for physics considerations

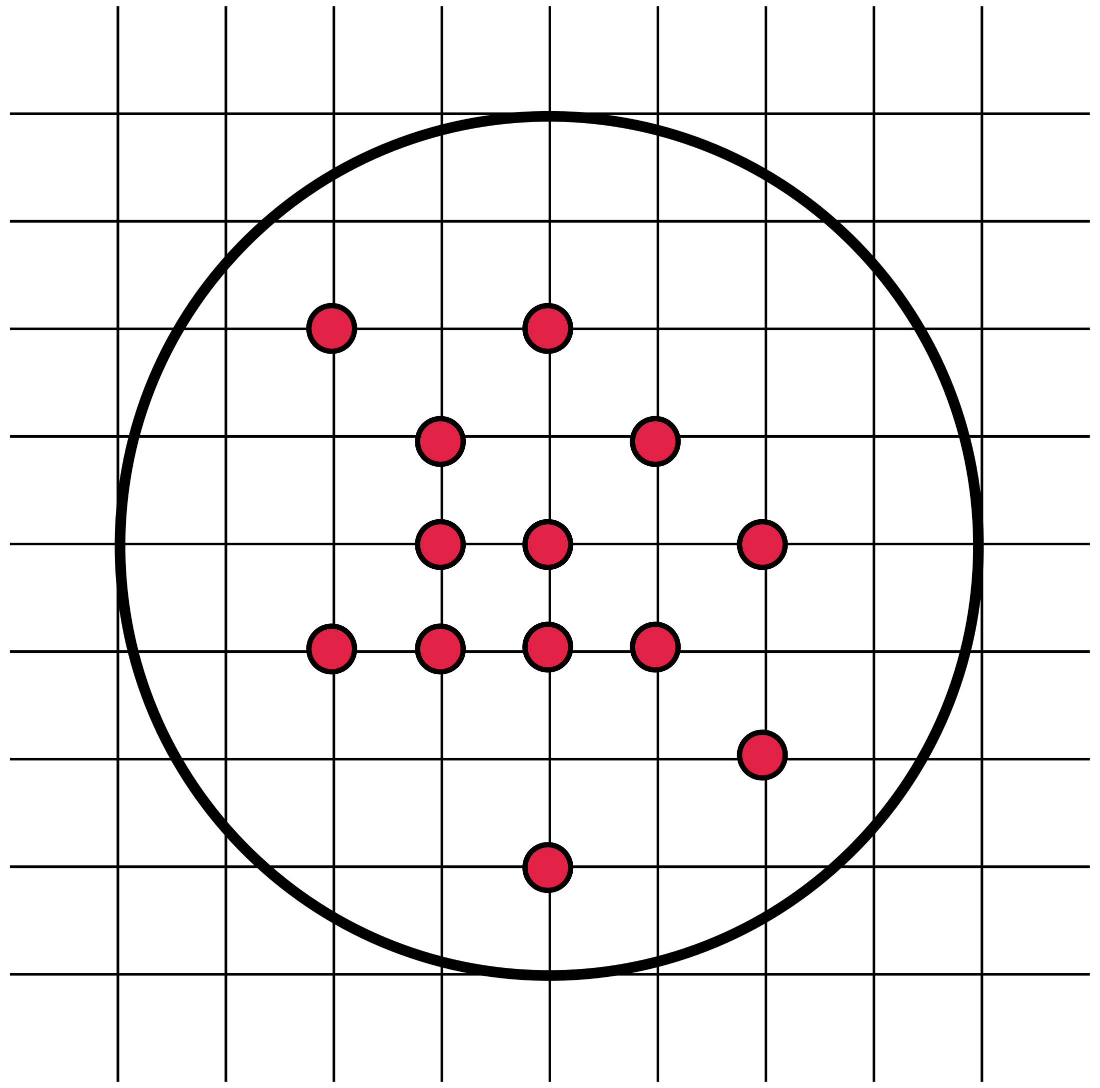
❶ Autoregressive model for $p(\boldsymbol{K})$

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

Wu, LW, Zhang, PRL '19

Fermionic occupation in k-space

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



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

$p(\textit{jumps}\,|\ldots)$

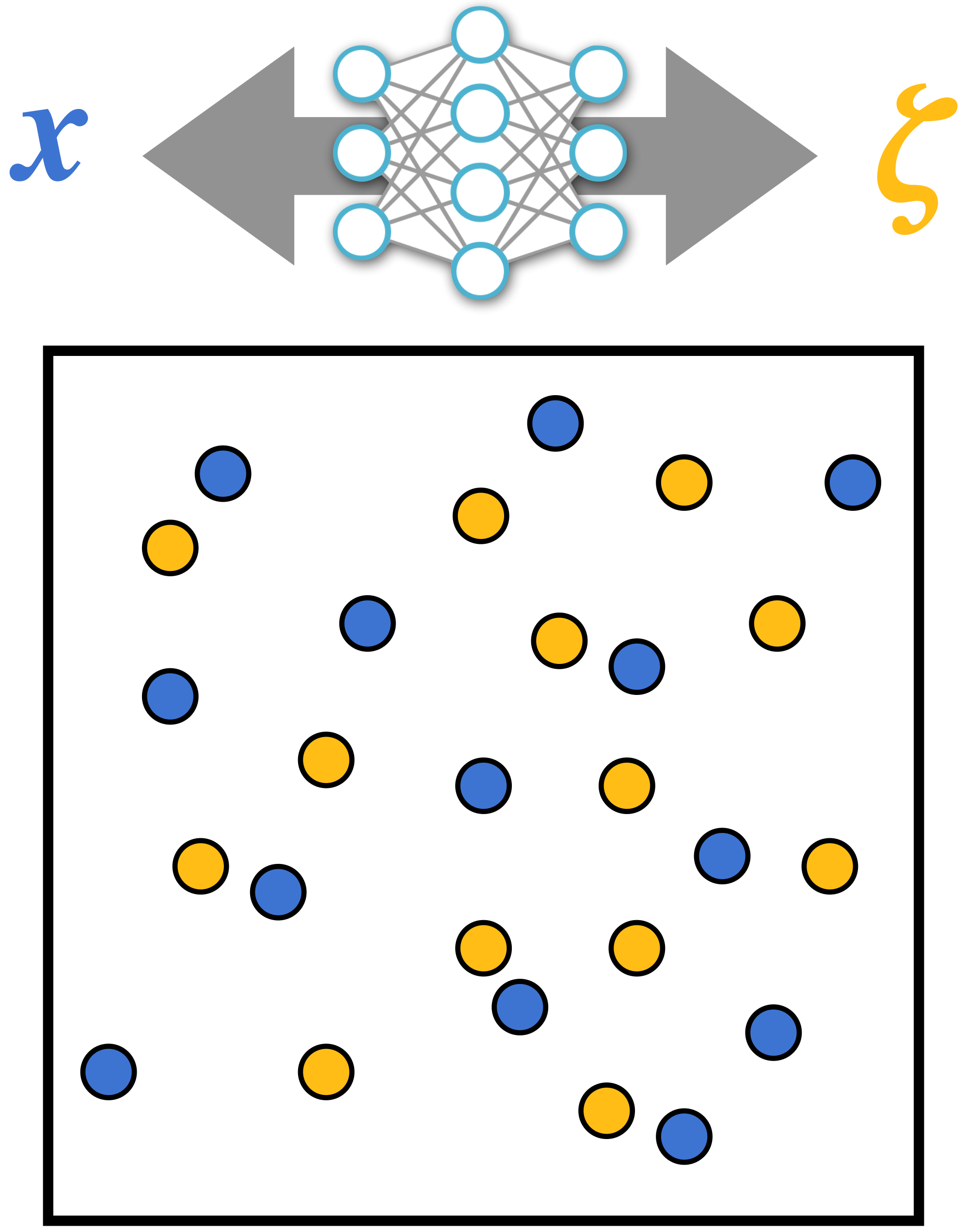
# of particles	# of words
Momentum cutoff	Vocabulary
Entropy	Negative log-likelihood

Twist: 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

② Normalizing flow for $|\Psi_{\boldsymbol{K}}\rangle$

Electron coordinates $\boldsymbol{x}$



ζ Quasi-particle coordinates

$$\Psi_{\boldsymbol{K}}(\boldsymbol{x}) = \frac{\det\left(e^{i\boldsymbol{k}_i\cdot\zeta_j}\right)}{\sqrt{N!}} \cdot \left|\det\left(\frac{\partial \zeta}{\partial \boldsymbol{x}}\right)\right|^{\frac{1}{2}}$$

Orthonormal many-body states

Jacobian of the transformation

Twist: the flow should be permutation equivariant for fermionic coordinates

we use FermiNet layer Pfau et al, 1909.02487

The objective function

$$F = \mathop{\mathbb{E}}_{\boldsymbol{K}\sim p(\boldsymbol{K})}\left[\frac{1}{\beta}\ln p(\boldsymbol{K}) + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\left|\langle \boldsymbol{x}|\Psi_{\boldsymbol{K}}\rangle\right|^2}\left[\frac{\langle \boldsymbol{x}|H|\Psi_{\boldsymbol{K}}\rangle}{\langle \boldsymbol{x}|\Psi_{\boldsymbol{K}}\rangle}\right]\right]$$

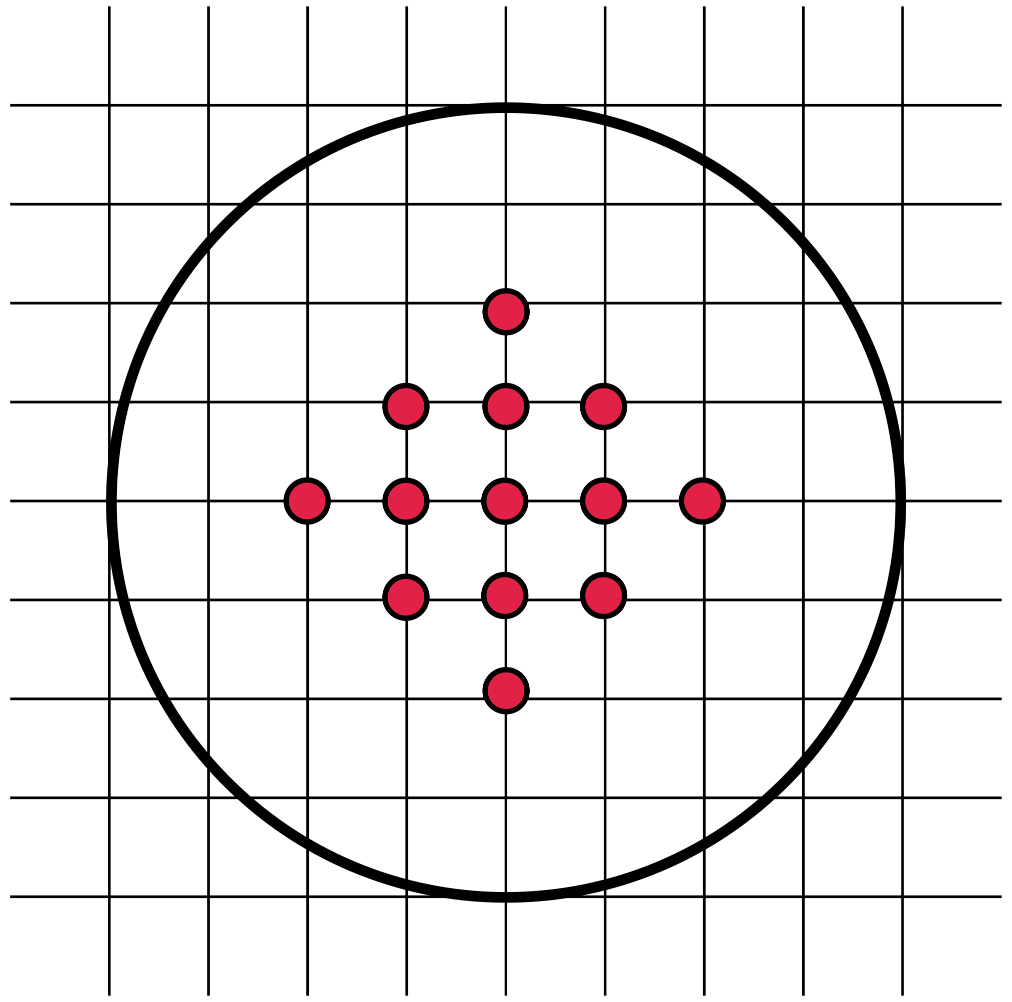
Boltzmann distribution

Born rule

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

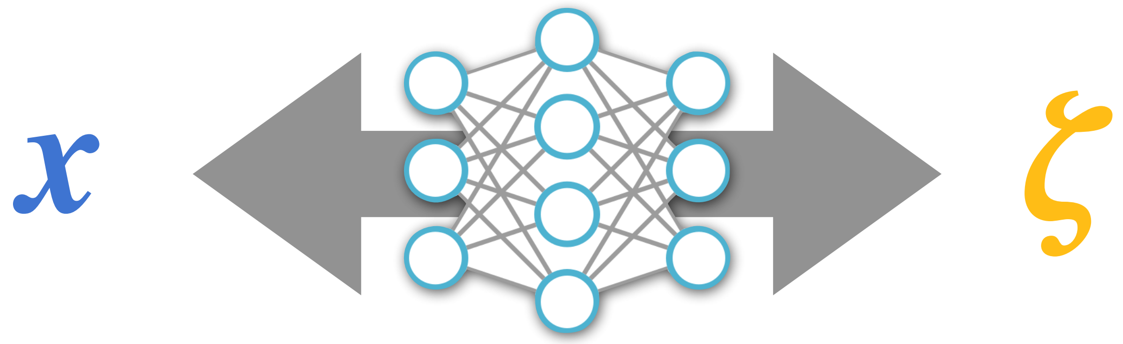
Limiting case 1: Interacting electrons at T=0

$\boldsymbol{K} = \{\boldsymbol{k}_i\}$



$p(\boldsymbol{K}) = 1$ only for the closed shell momentum configuration

$$E = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim |\Psi_{\boldsymbol{K}}(\boldsymbol{x})|^2} \left[\frac{\langle \boldsymbol{x} | H | \Psi_{\boldsymbol{K}} \rangle}{\langle \boldsymbol{x} | \Psi_{\boldsymbol{K}} \rangle} \right]$$



Reduces to ground state variational Monte Carlo with a single normalizing flow wavefunction

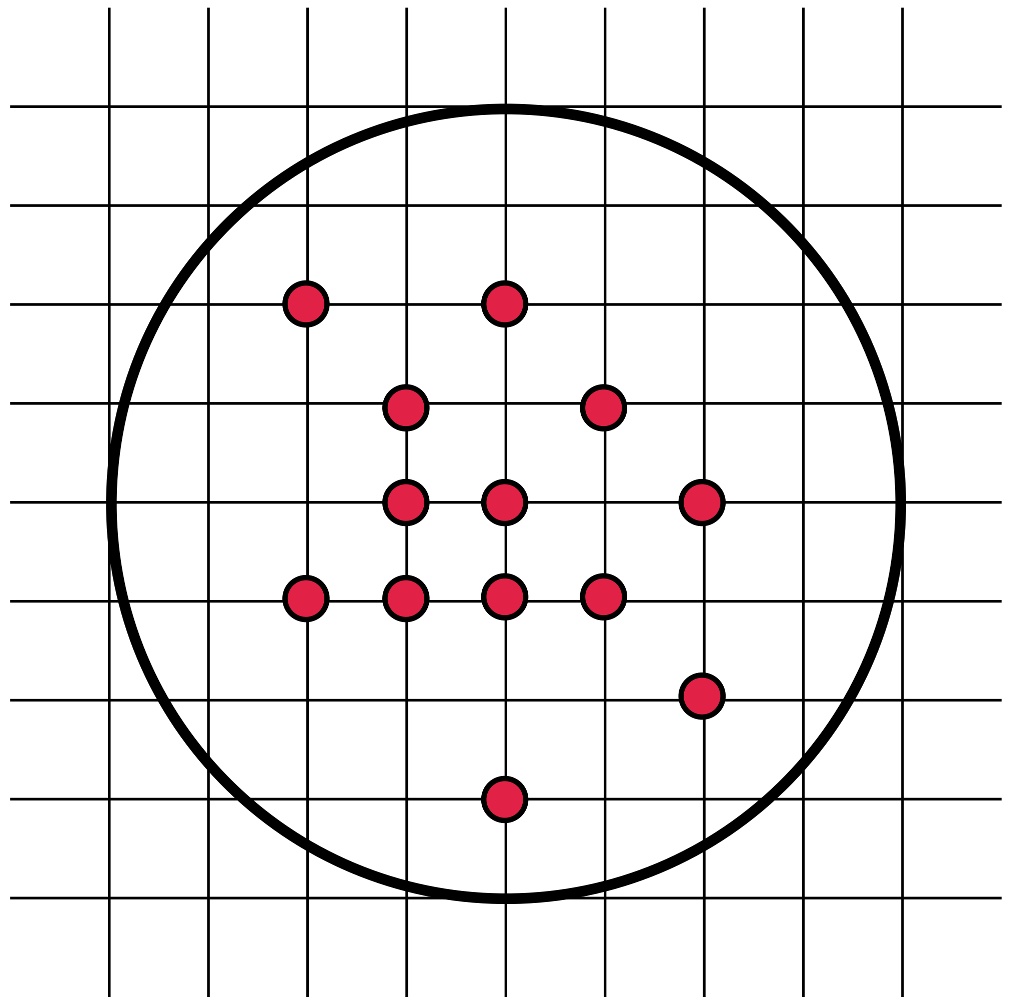
c.f. neural network states for uniform electron gases: Wilson et al 2202.04622, Cassella et al 2202.05183 , Li et al, 2203.15472

Limiting case 2: Noninteracting electrons at T>0

$\boldsymbol{R} = \zeta$

$$F = \mathop{\mathbb{E}}_{K \sim p(\boldsymbol{K})} \left[\frac{1}{\beta} \ln p(\boldsymbol{K}) + \sum_{i=1}^{N} \frac{\hbar^2 \boldsymbol{k}_i^2}{2m} \right]$$

$\boldsymbol{K} = \{\boldsymbol{k}_i\}$



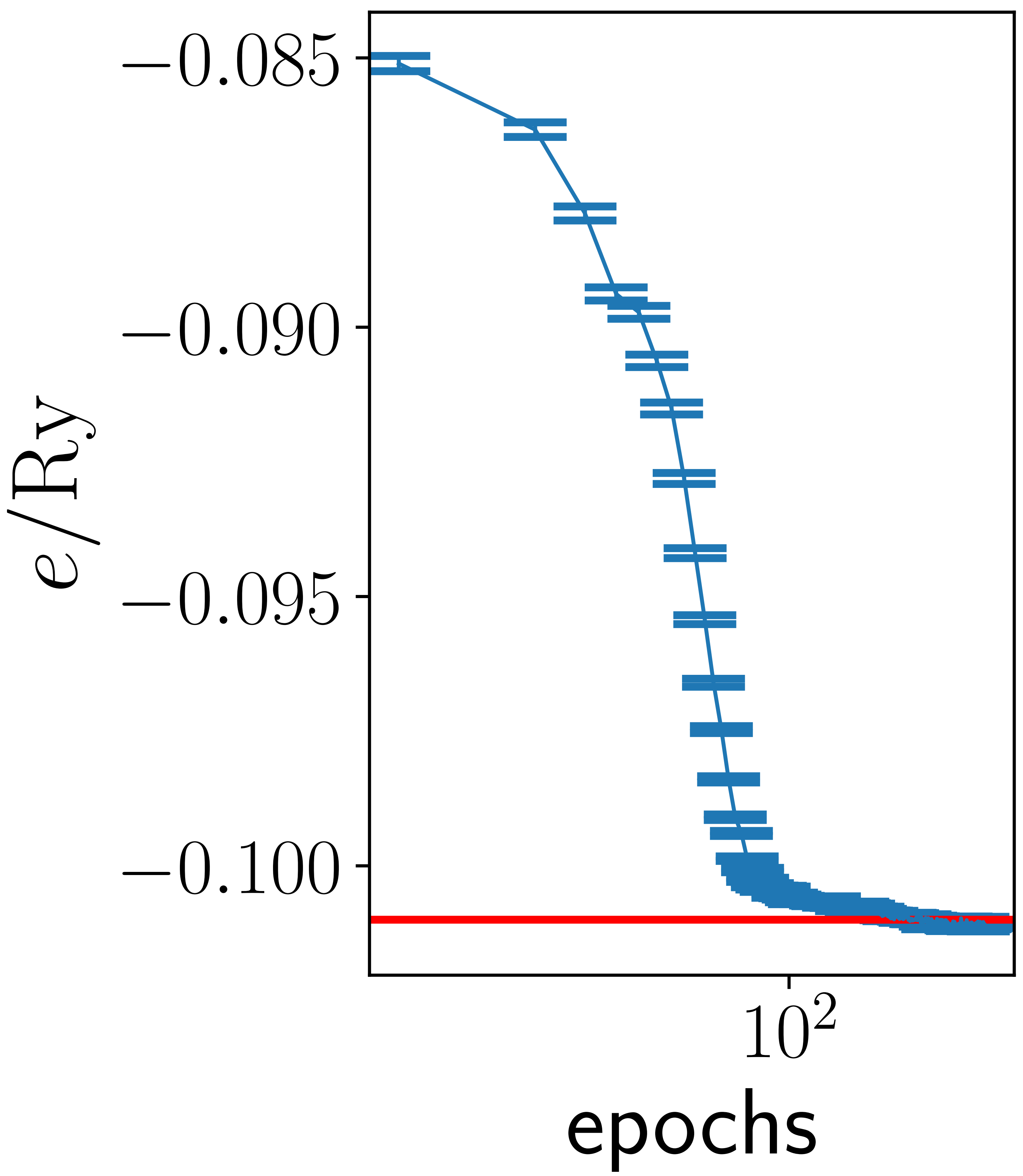
A classical statistical mechanics problem:
Noninteracting fermions in canonical ensemble

(Not as trivial as you might think) Borrmann & Franke, J. Chem. Phys. 1993

Distribute N fermions in M momenta to minimize the free energy

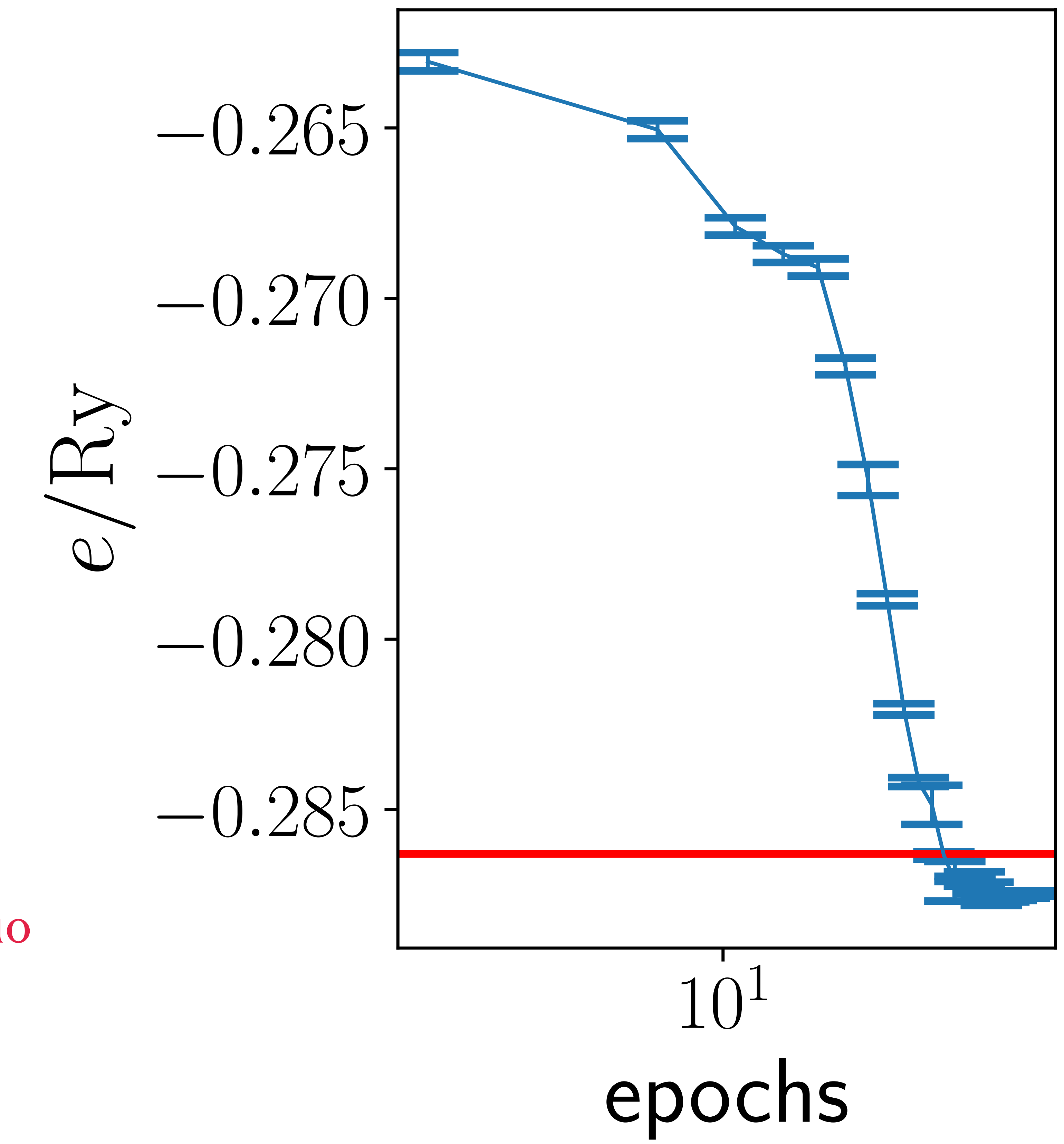
Benchmarks on spin-polarized electron gases

3D electron gas T/T_F=0.0625



Brown et al, PRL '13
restricted PIMC N=33, r_s=10

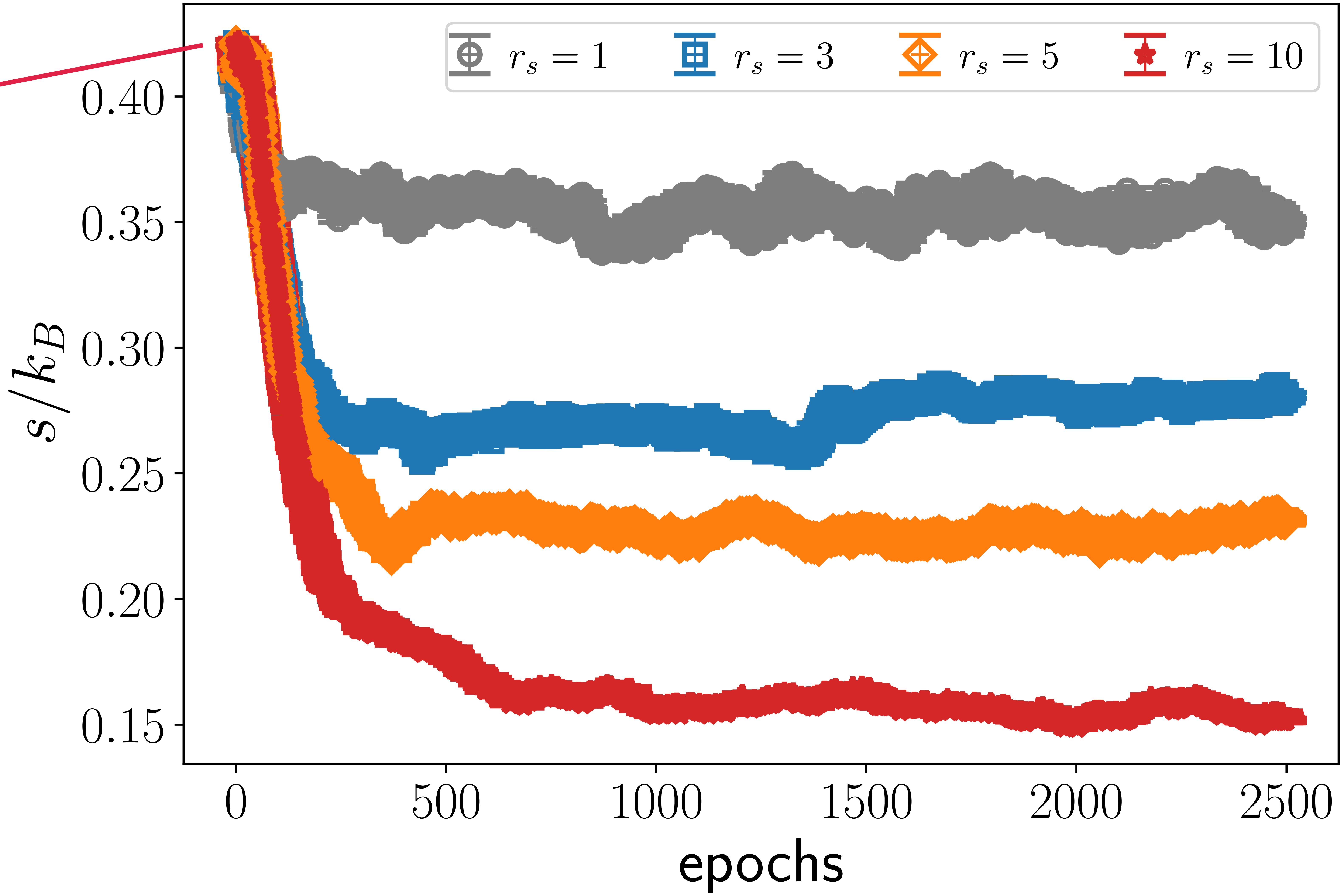
2D electron gas T=0



Tanatar, Ceperley, PRB, '89
Slater-Jastrow VMC N=37, r_s=5

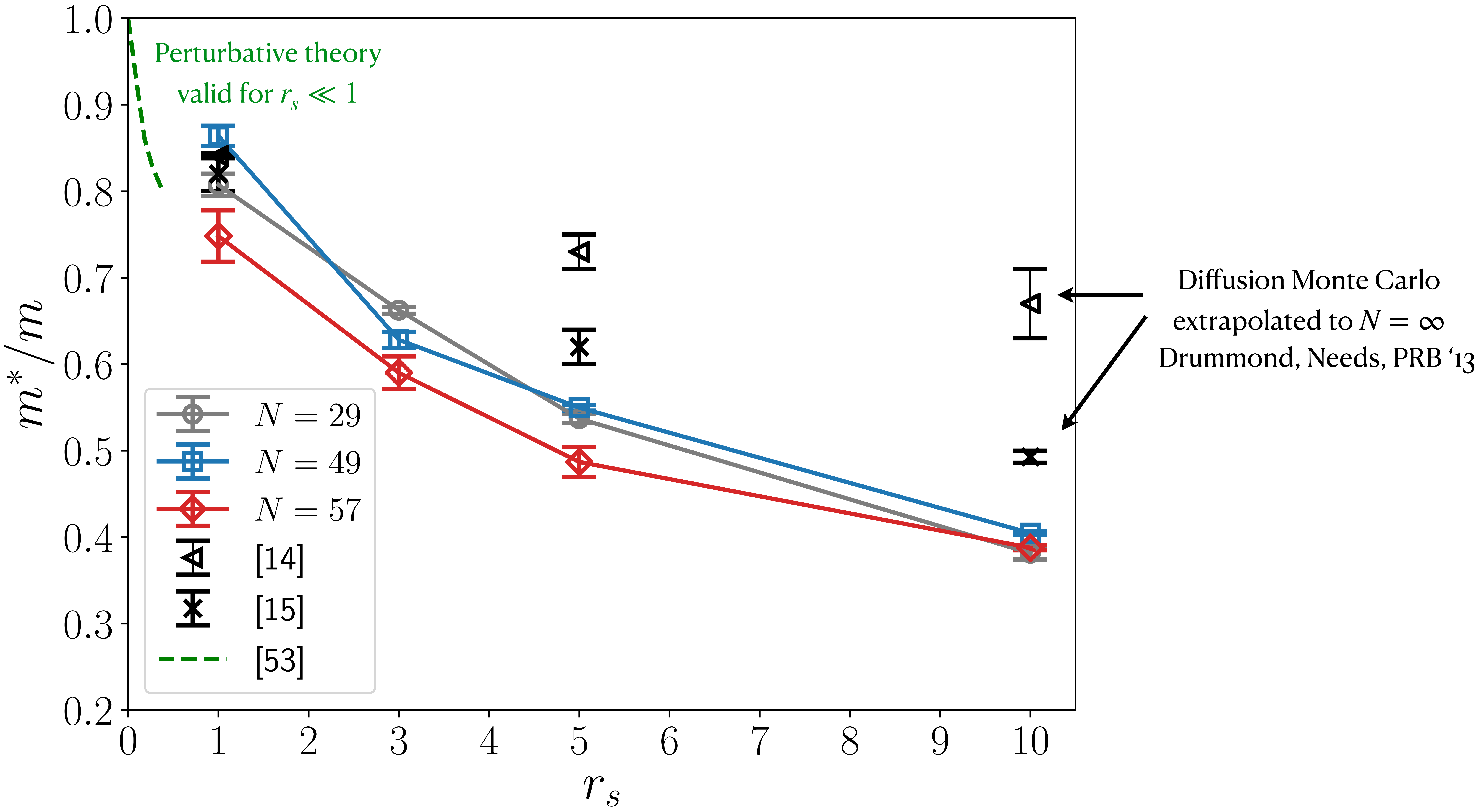
37 spin-polarized electrons in 2D @ T/T_F=0.15

Starting point:
Ideal gas entropy s_0



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

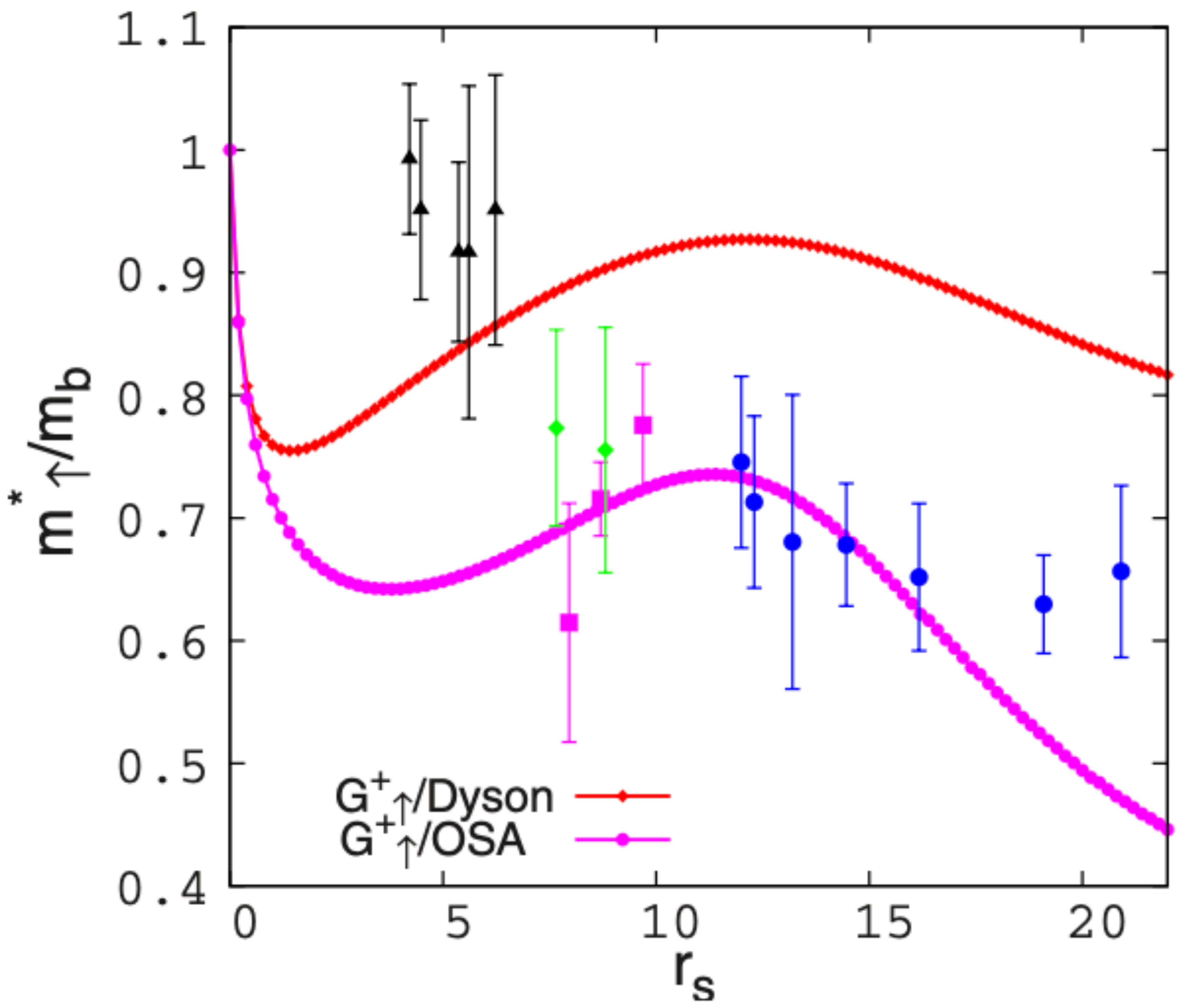
Effective mass of spin-polarized 2DEG



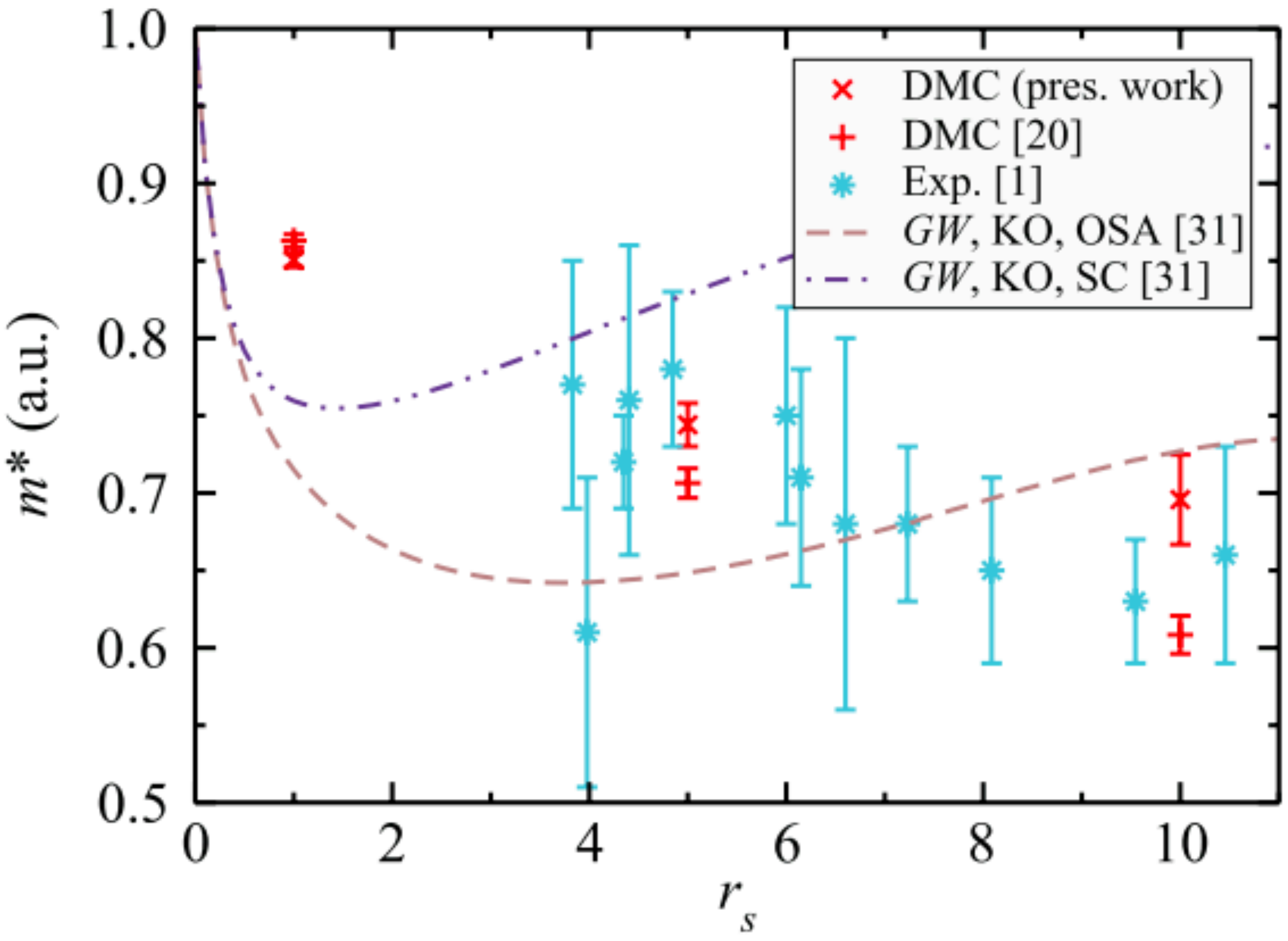
More pronounced suppression of m^* in the low-density strong-coupling region

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

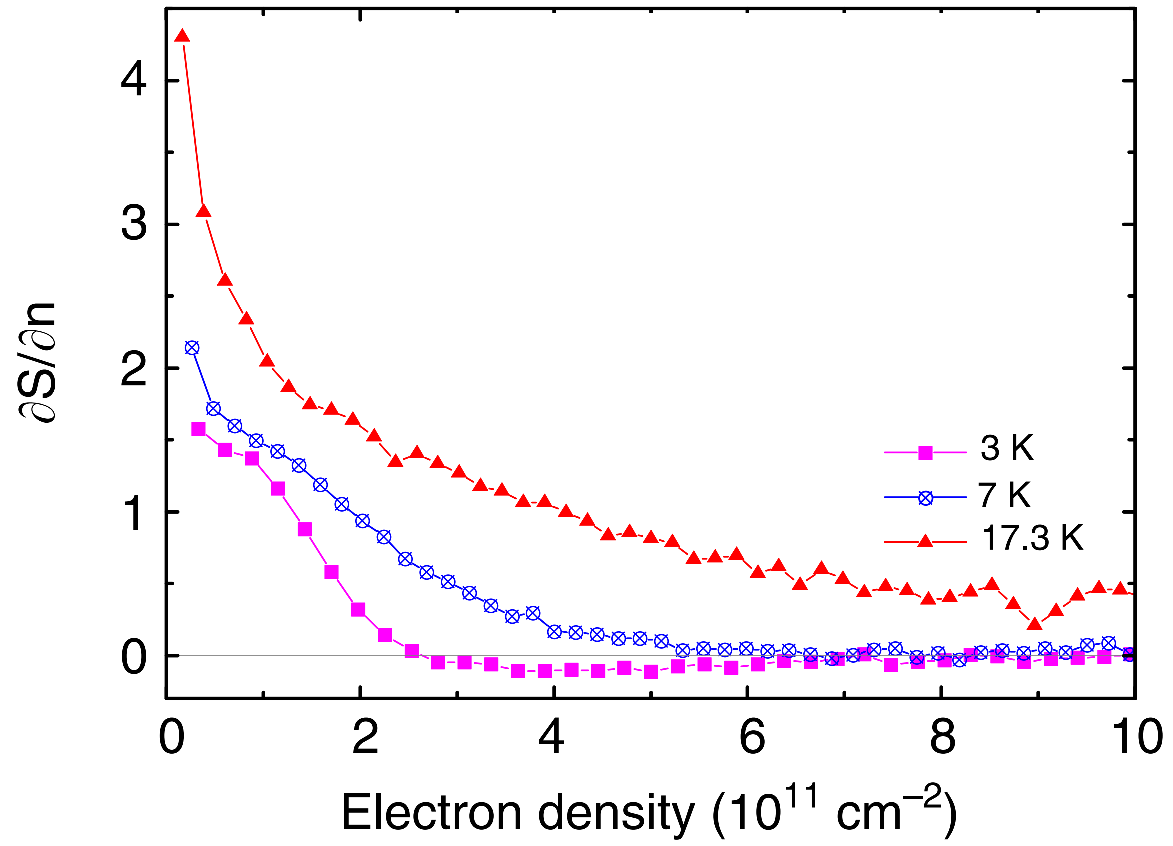
Received 16 May 2014 | Accepted 27 Apr 2015 | Published 23 Jun 2015

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

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 machine learning

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

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

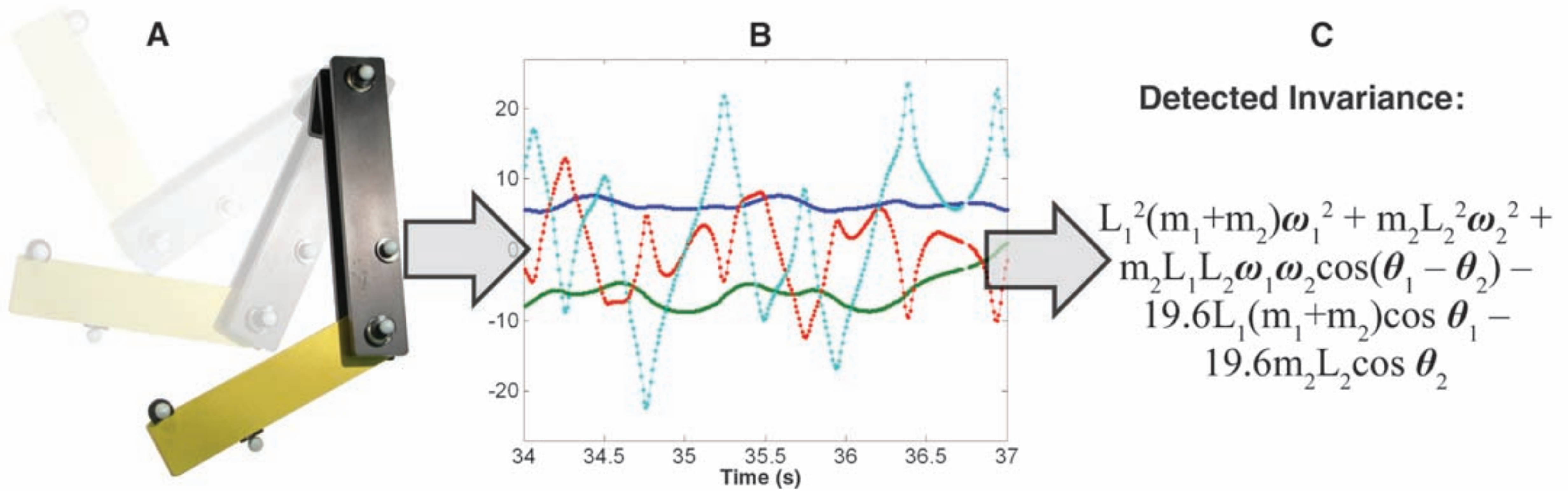
$\zeta \leftrightarrow \boldsymbol{x}$ illustrates adiabatic continuity.

Discussions

Can machines discover physical law ?

Distilling Free-Form Natural Laws from Experimental Data

Schmidt, Lipson
Science '09



References and Notes

1. P. W. Anderson, *Science* **177**, 393 (1972).
2. E. Noether, *Nachr. d. König Gesellsch. d. Wiss. zu Göttingen, Math-Phys. Klasse* 235 (1918).

Machines Fall Short of Revolutionary Science

In the Report by Schmidt and Lipson, a machine deduces the equation behind a sample of chaotic motion. The discovery of deterministic chaos is an example of true Kuhnian revolution; others were its application to unexpected fields like meteorology and population biology. In the constrained problem in the Report, the relevant physical law and variables are known in advance; it is hardly a template for the creative, exploratory nature of true science.

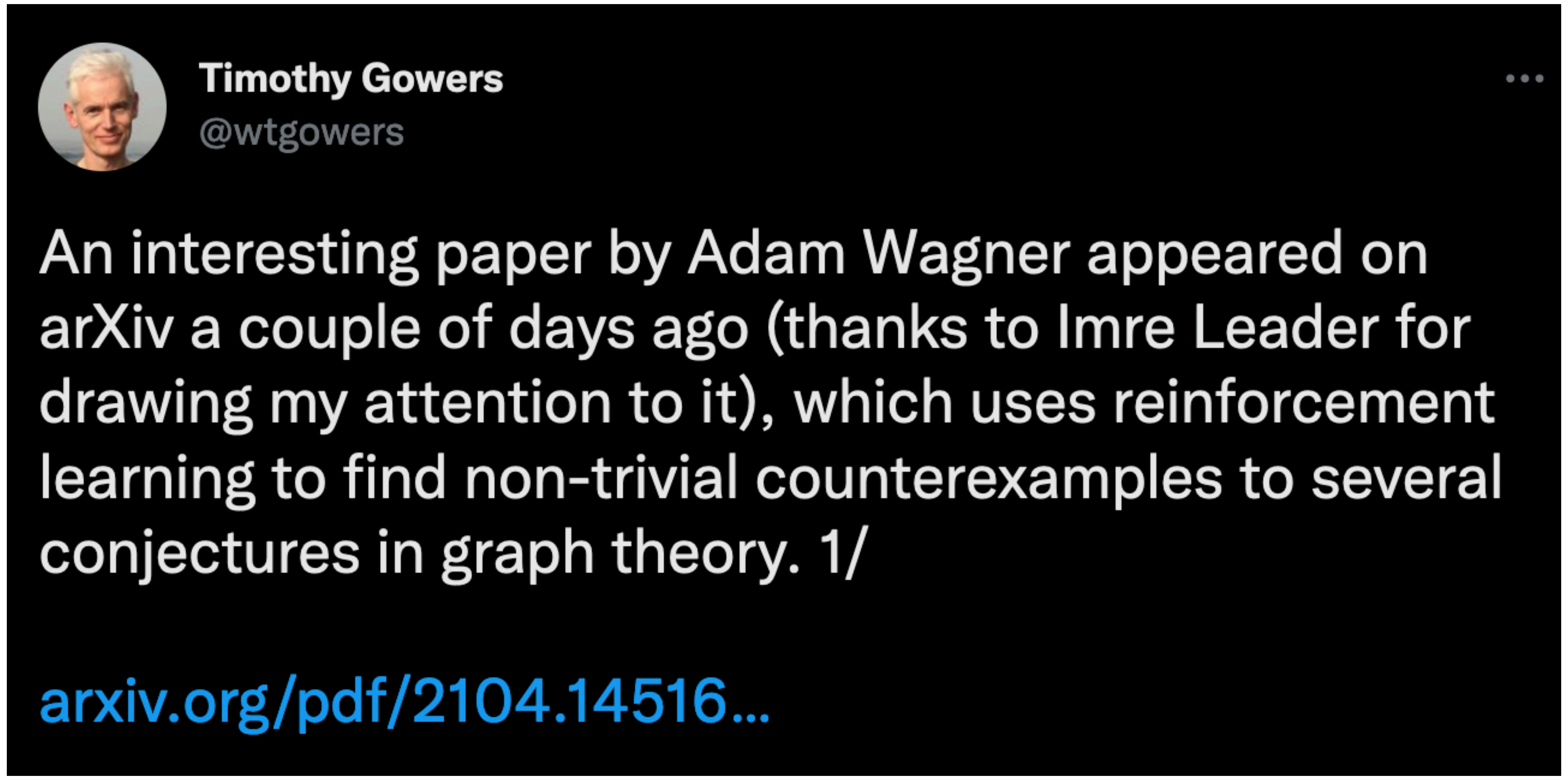
PHILIP W. ANDERSON[1*] AND ELIHU ABRAHAMS[2]

汤超院士在2022科学智能峰会上的讲话：“关于AI for Science的几层意思”

https://mp.weixin.qq.com/s/oL7G7ByazbnsgrXDToPyrw

Discussions

Can machines discover mathematics?


Timothy Gowers
@wtgowers

An interesting paper by Adam Wagner appeared on arXiv a couple of days ago (thanks to Imre Leader for drawing my attention to it), which uses reinforcement learning to find non-trivial counterexamples to several conjectures in graph theory. 1/

arxiv.org/pdf/2104.14516...

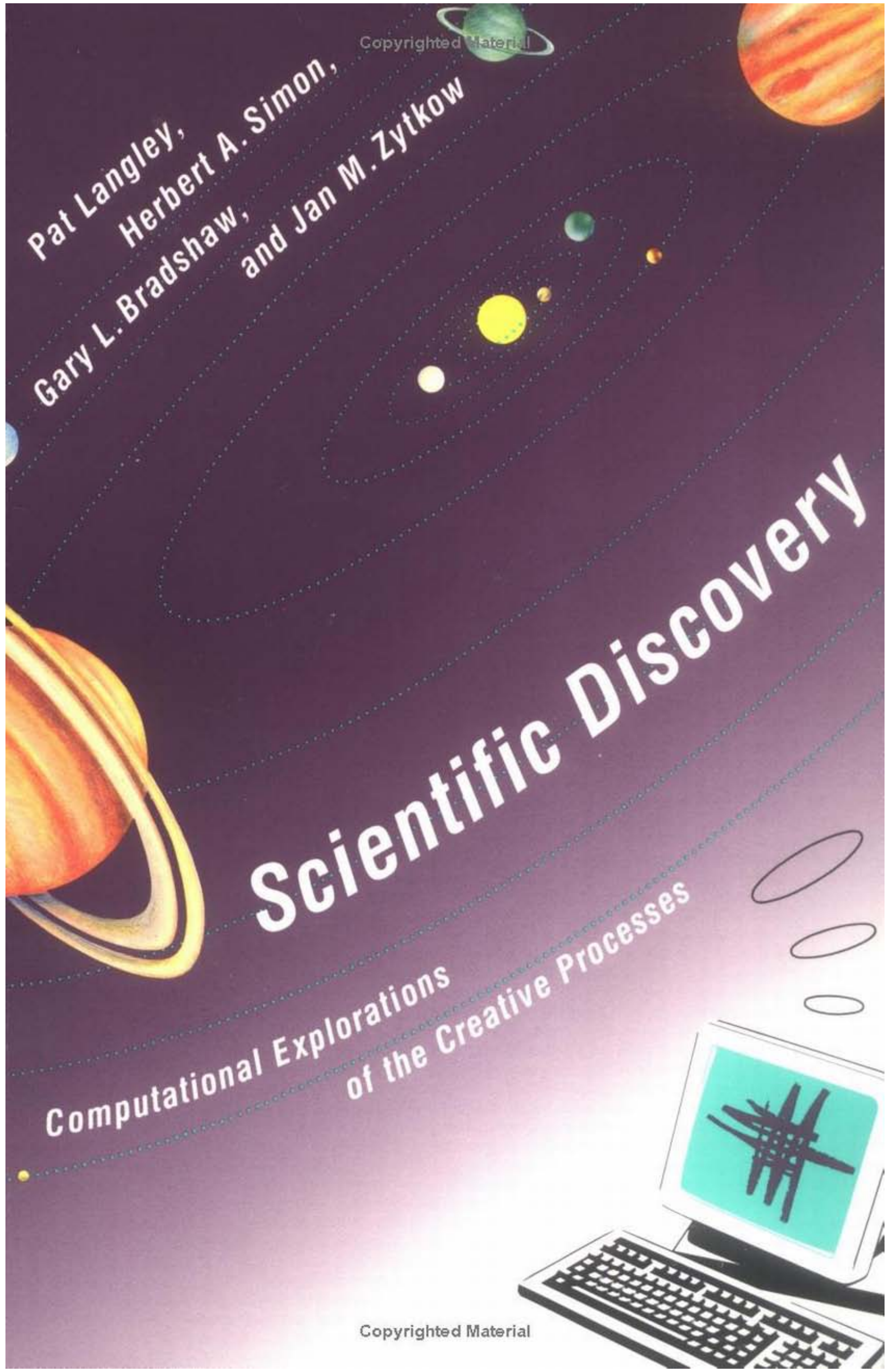
Nature 600, 70 (2021)

Advancing mathematics by guiding human intuition with AI

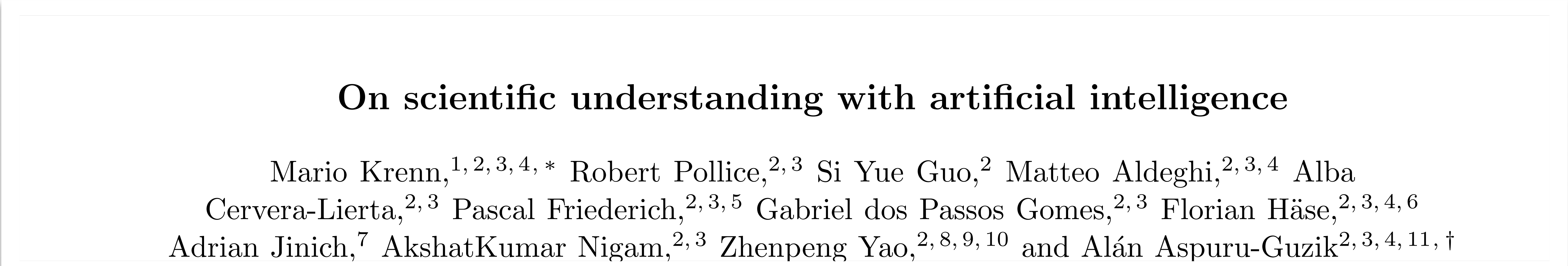
Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis & Pushmeet Kohli

Search counter-examples to reject conjectures

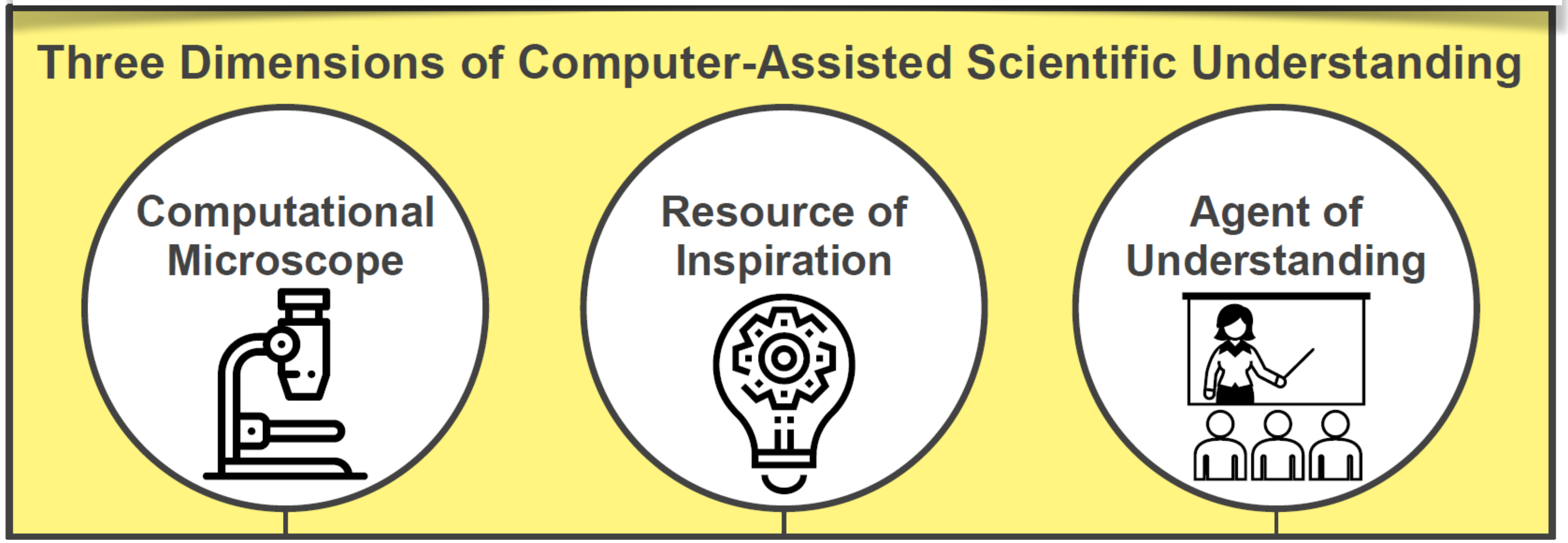
Guide human mathematician to propose conjectures



2204.01467

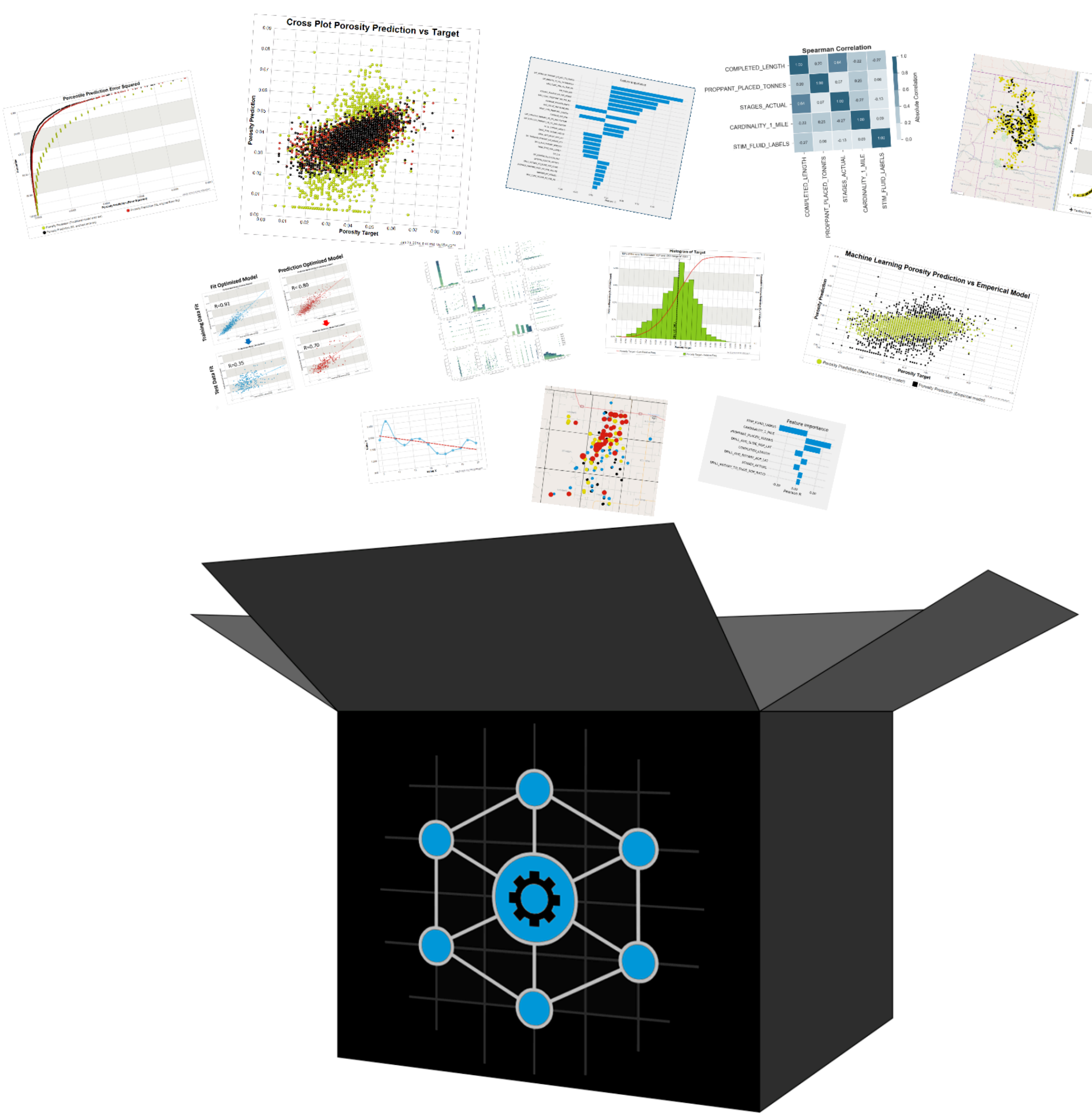

On scientific understanding with artificial intelligence

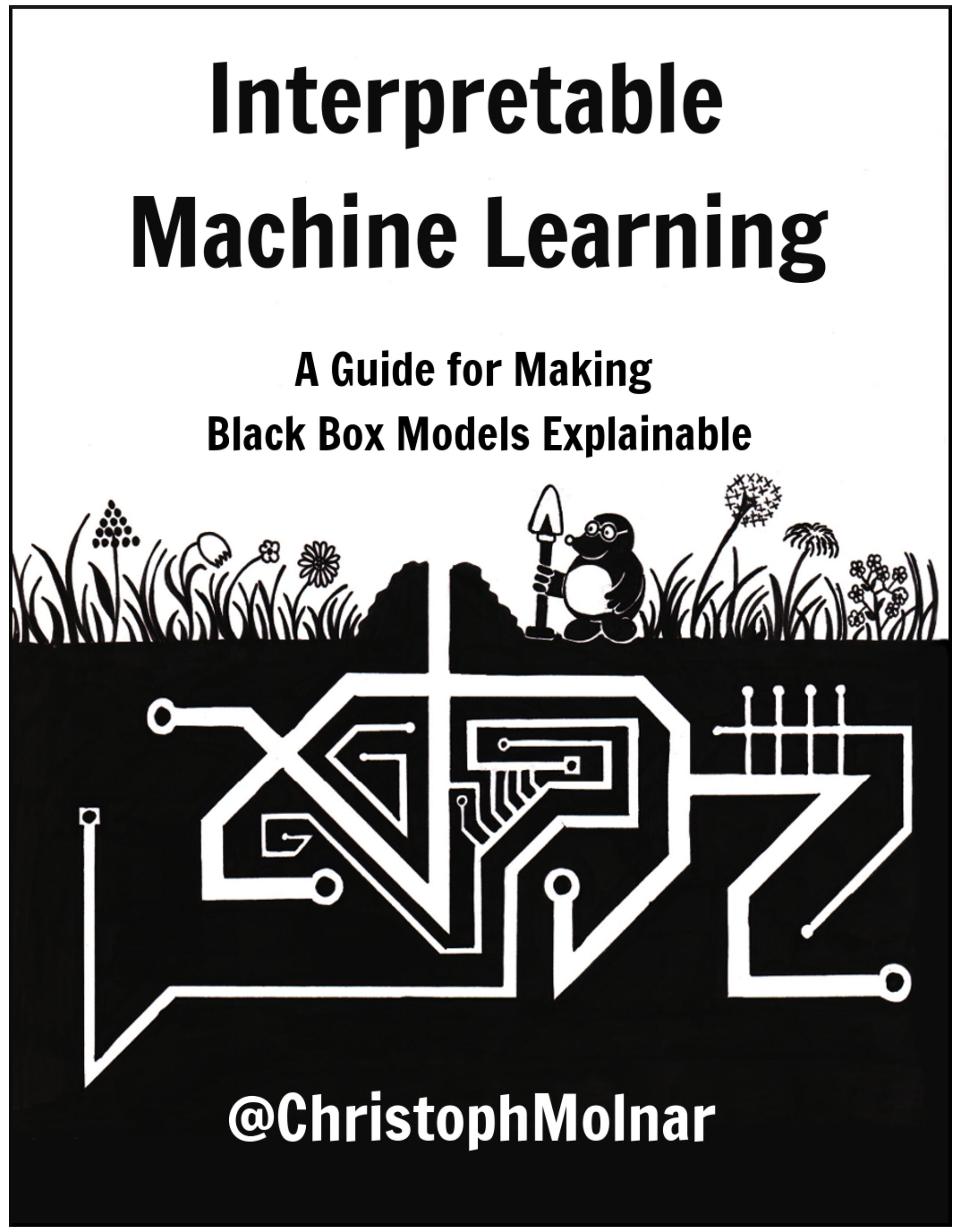
Mario Krenn,[1,2,3,4,*] Robert Pollice,[2,3] Si Yue Guo,[2] Matteo Aldeghi,[2,3,4] Alba Cervera-Lierta,[2,3] Pascal Friederich,[2,3,5] Gabriel dos Passos Gomes,[2,3] Florian Häse,[2,3,4,6] Adrian Jinich,[7] AkshatKumar Nigam,[2,3] Zhenpeng Yao,[2,8,9,10] and Alán Aspuru-Guzik[2,3,4,11,†]



Discussions

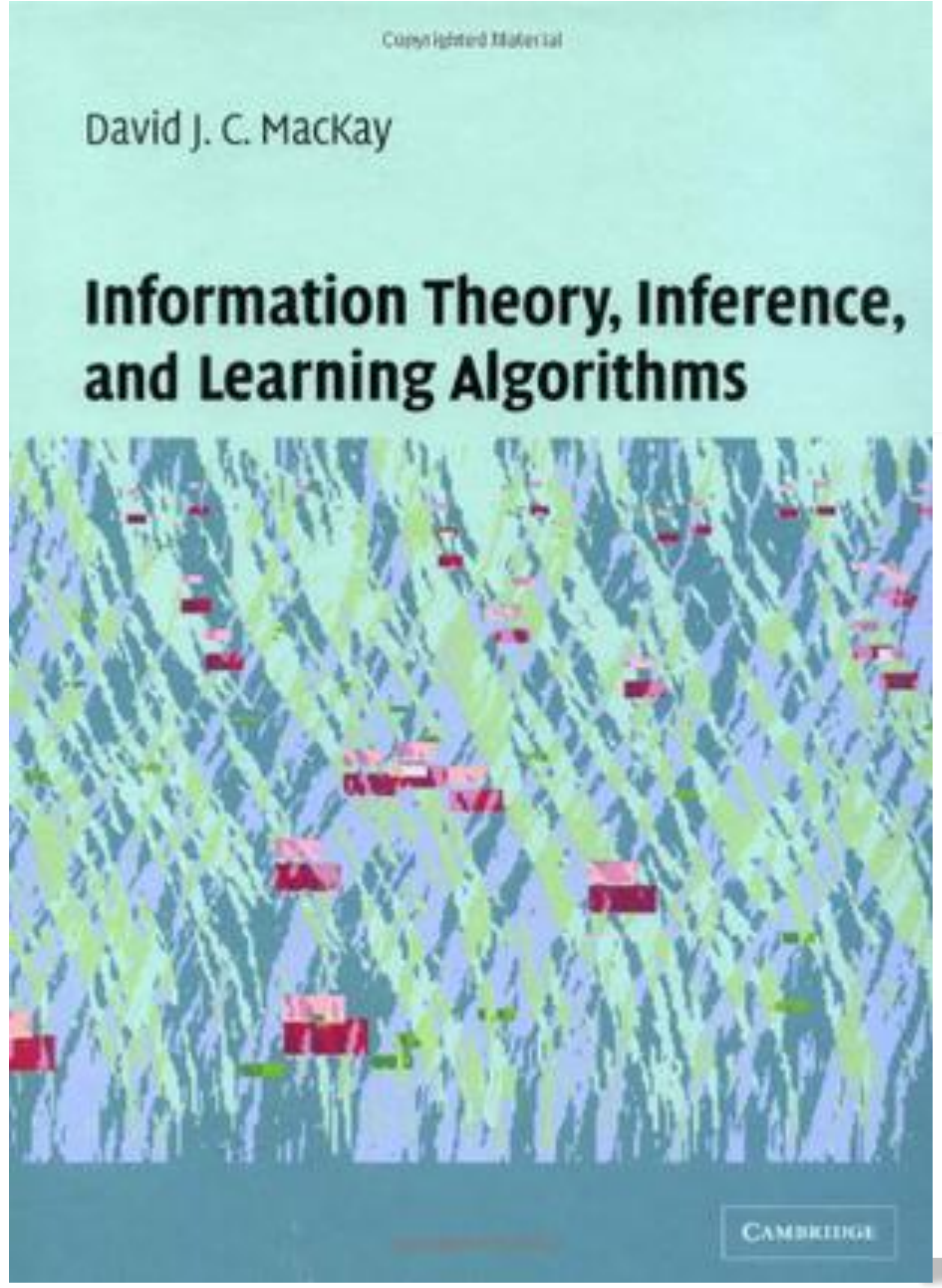
Do we understand what is the machine doing ?





Yes/No/Well, do I have to ?/I don't care...

Discussions



Is this all fitting ?

One of my students, Robert, asked:

> Maybe I'm missing something fundamental, but supervised neural networks seem equivalent to fitting a pre-defined function to some given data, then extrapolating – what's the difference?

I agree with Robert. The supervised neural networks we have studied so far are simply parameterized nonlinear functions which can be fitted to data.

True for supervised learning, which is hugely successful for real-world applications.
But that is not the whole story, especially for scientific applications.

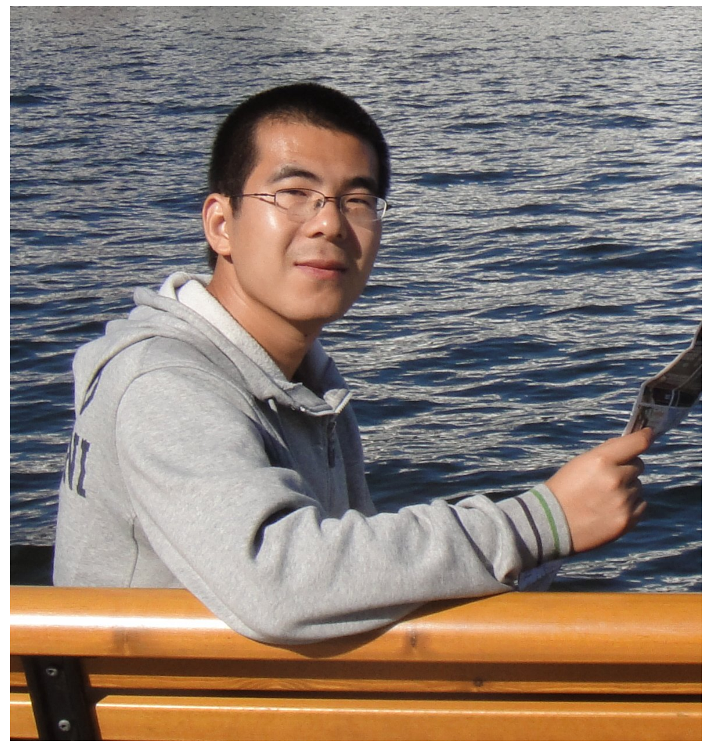
“Using AI to accelerate scientific discovery” 2021, by Demis Hassabis, co-founder and CEO of DeepMind

Thank you!

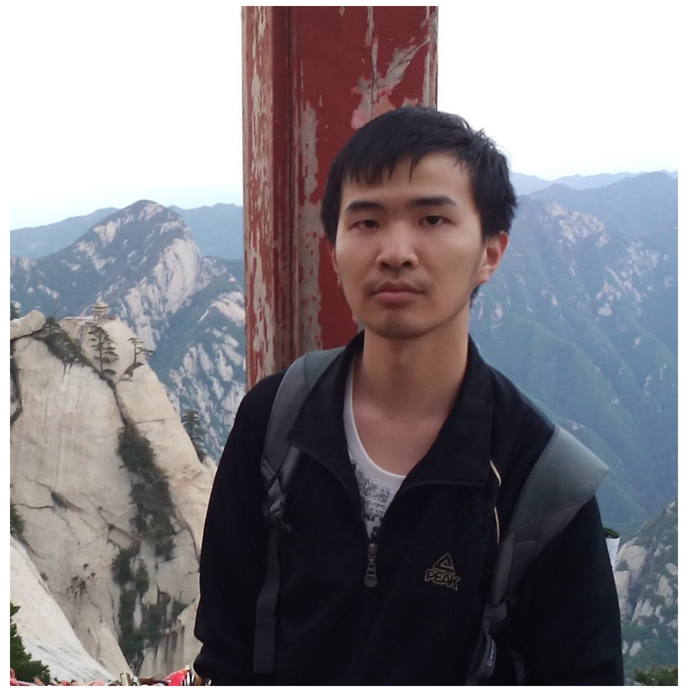
Deep generative model-based variational free-energy calculations

Shuo-Hui Li

Dian Wu



Pan Zhang



Hao Xie

Linfeng Zhang

1802.02840, PRL '18
1809.10606, PRL '19
2105.08644, JML '22
2201.03156

[li012589/NeuralRG]
[wdphy16/stat-mech-van]
[fermiflow/fermiflow]
[fermiflow/CoulombGas]