Generative AI for Science

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

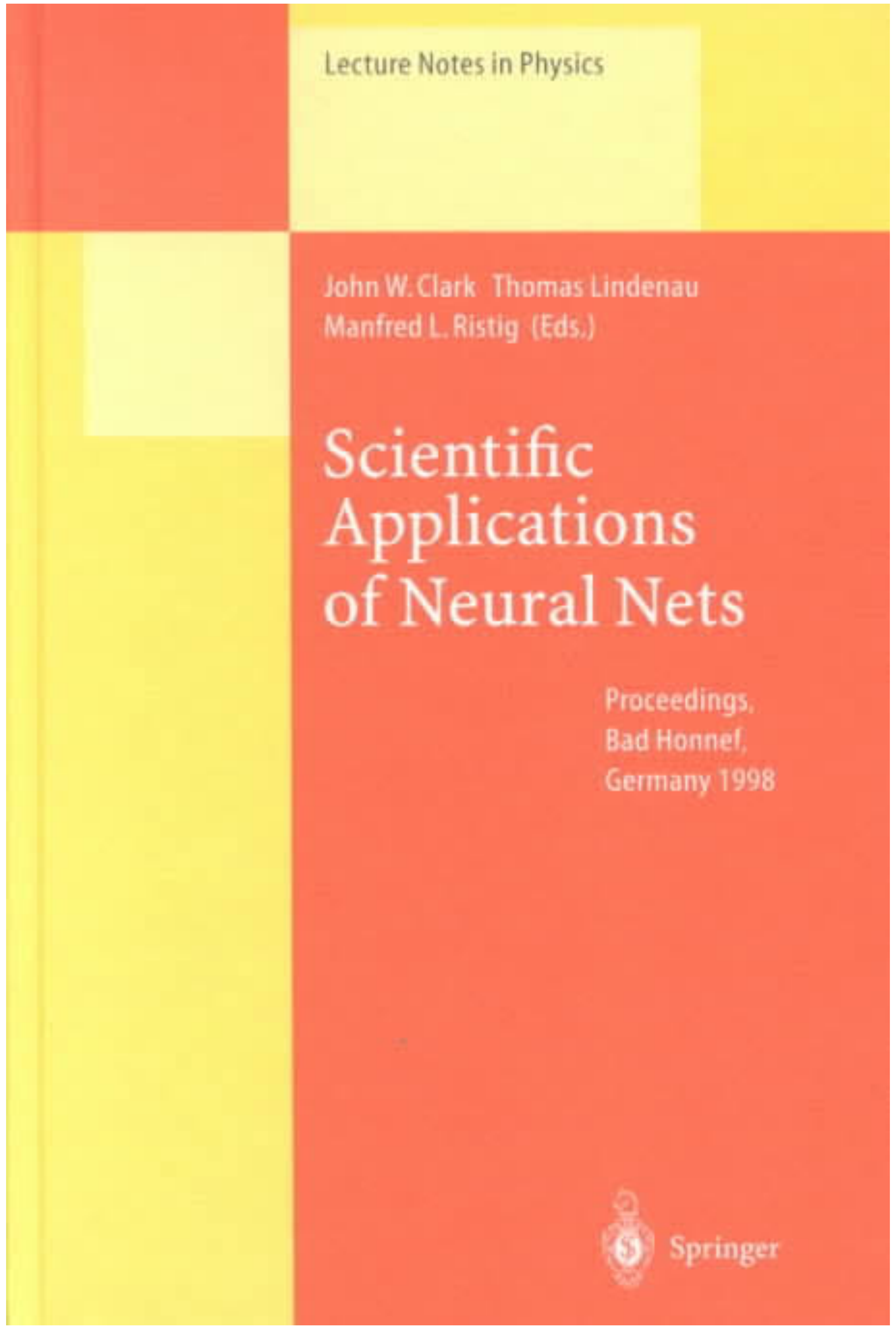
Plan

1. Motivation: AI for science, why now ?

2. Generative models and their physics genes

3. Applications: electron gases and dense hydrogen

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

⋮

In conclusion, as a methodology for classification or function approximation in scientific problems, computational analysis based on neural networks is expected to prove most valuable in applications for which (i) the data set is large and complex, (ii) there is as yet no coherent theory of the underlying phenomenon, or quantitative theoretical explication is impractical,

Why now, again ?

What has changed ?

What has not ?

Science is more than fitting, so is machine learning

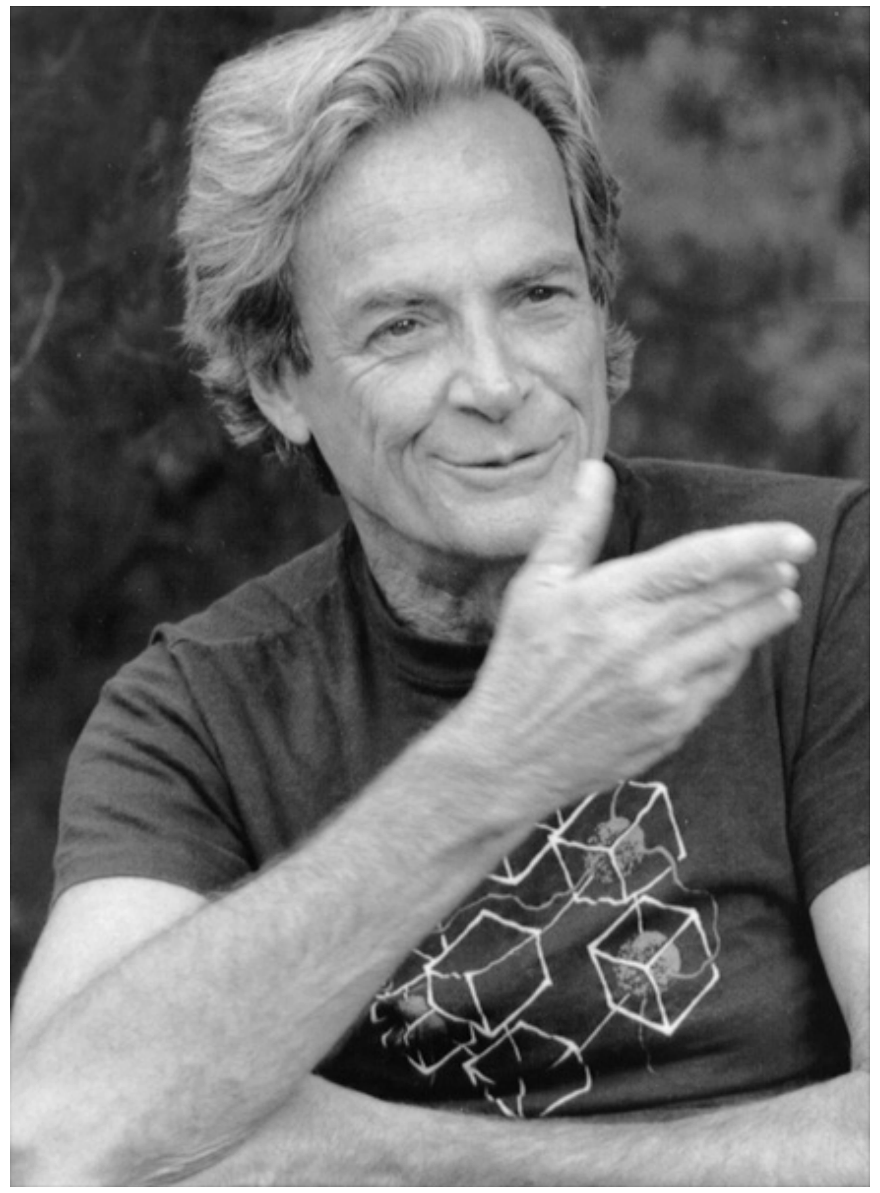
Discriminative learning

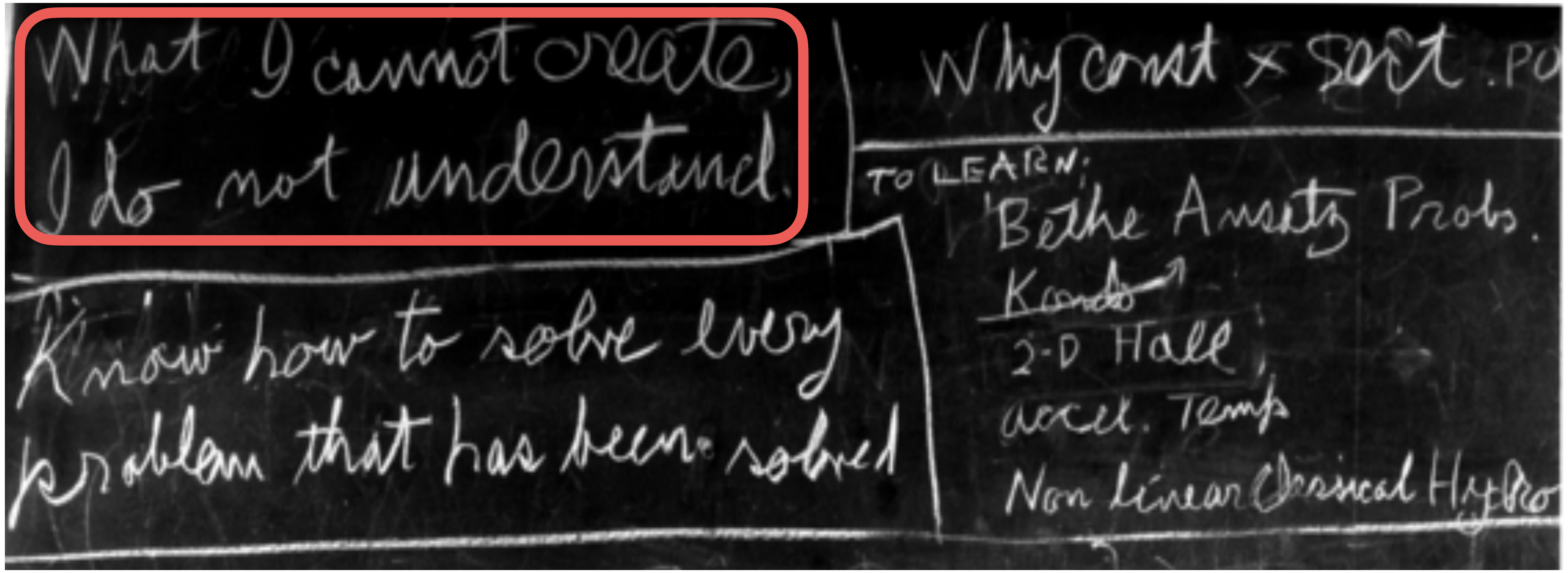
$$y = f(\boldsymbol{x})$$

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

Generative learning

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





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

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

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

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

Introducing Whisper
September 21, 2022 — Research

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

Our Approach to Alignment Research
August 24, 2022 — Research

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

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

OpenAI Technical Goals
June 20, 2016 — Announcements

Generative Models
June 16, 2016 — Research, Milestones

Team Update
May 25, 2016 — Announcements

OpenAI Gym Beta
April 27, 2016 — Research

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

Team++
March 31, 2016 — Announcements

Introducing OpenAI
December 11, 2015 — Announcements

https://openai.com/blog/

Generative AI: a new buzz word in silicon valley

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

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

New York Times
Kevin Roose
Oct. 21, 2022

Protocol
Biz Carson
October 21, 2022

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

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

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

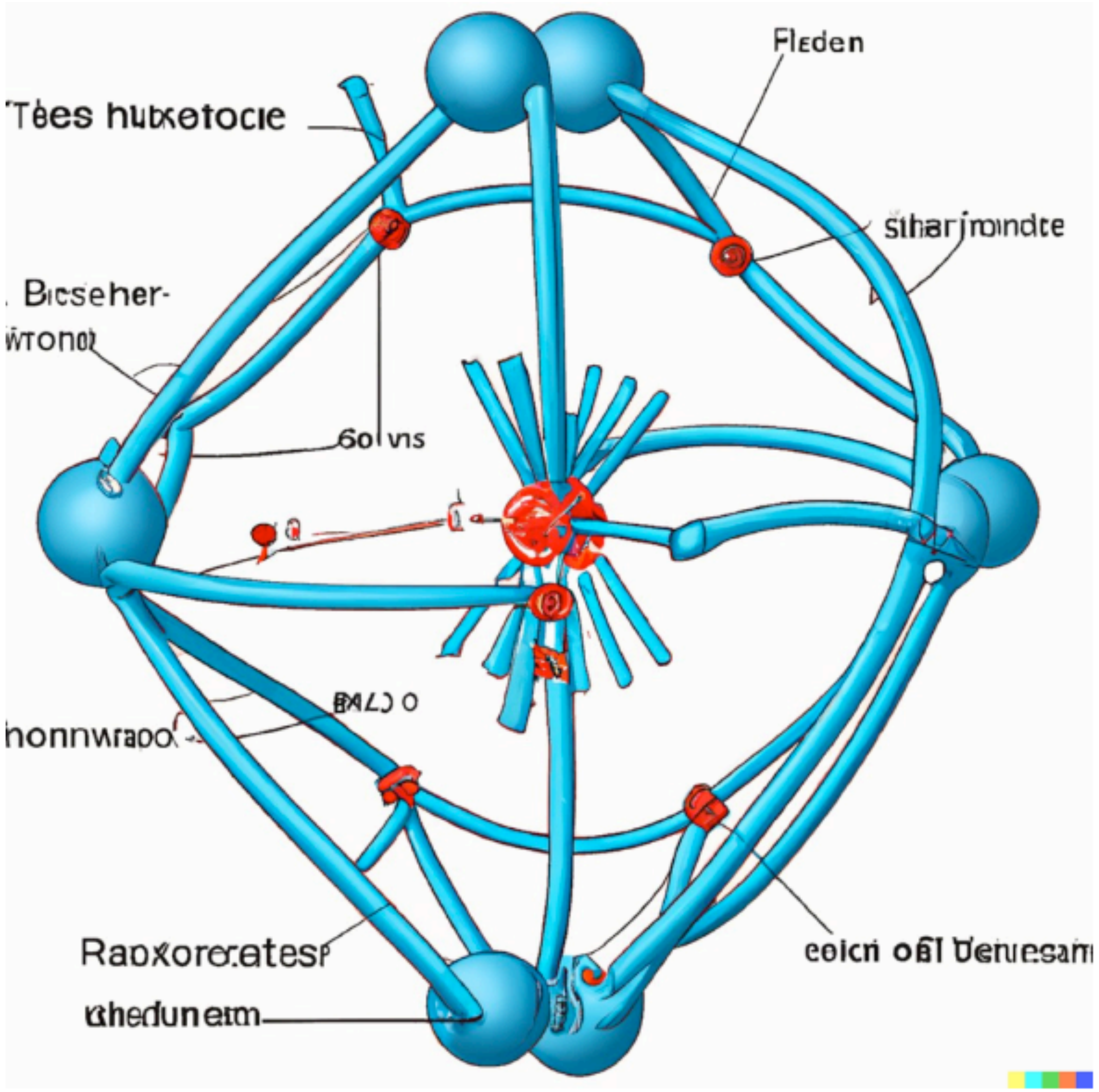
by Sonya Huang, Pat Grady and GPT-3

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

Large model availability: First attempts | Almost there | Ready for prime time https://huggingface.co/spaces/stabilityai/stable-diffusion

the inner structure of an electron

Generate image



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

How to Build a GPT-3 for Science (scientific literature and data)

Generative **P**re- **T**raining

Josh Nicholson

Posted August 18, 2022

You may ask (prompts):

"Tell me why this hypothesis is wrong"

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

"Generate a new treatment idea"

"What evidence is there to support social policy X?"

"Who has published the most reliable research in this field?"

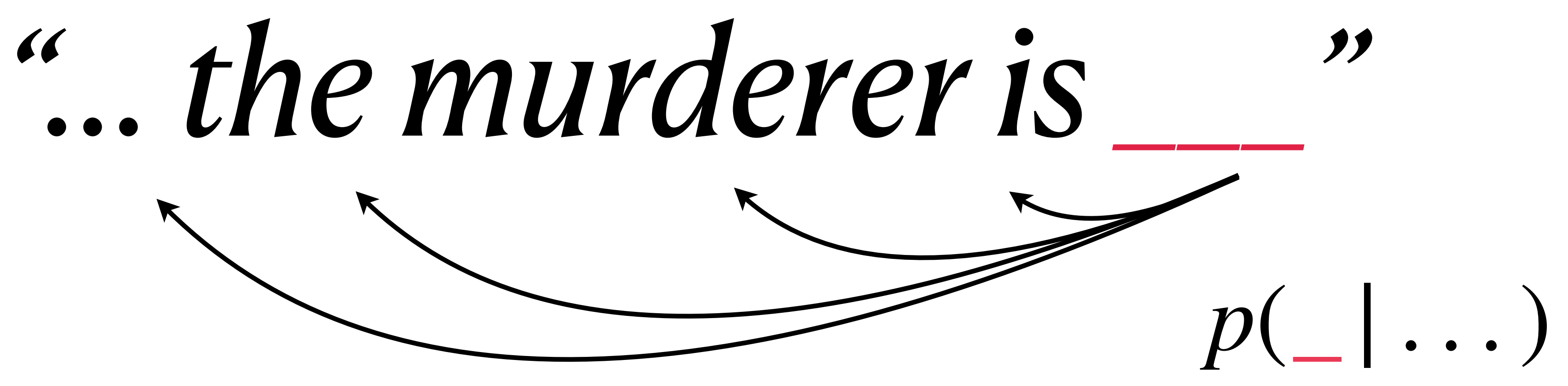
"Write me a scientific paper based on my data"

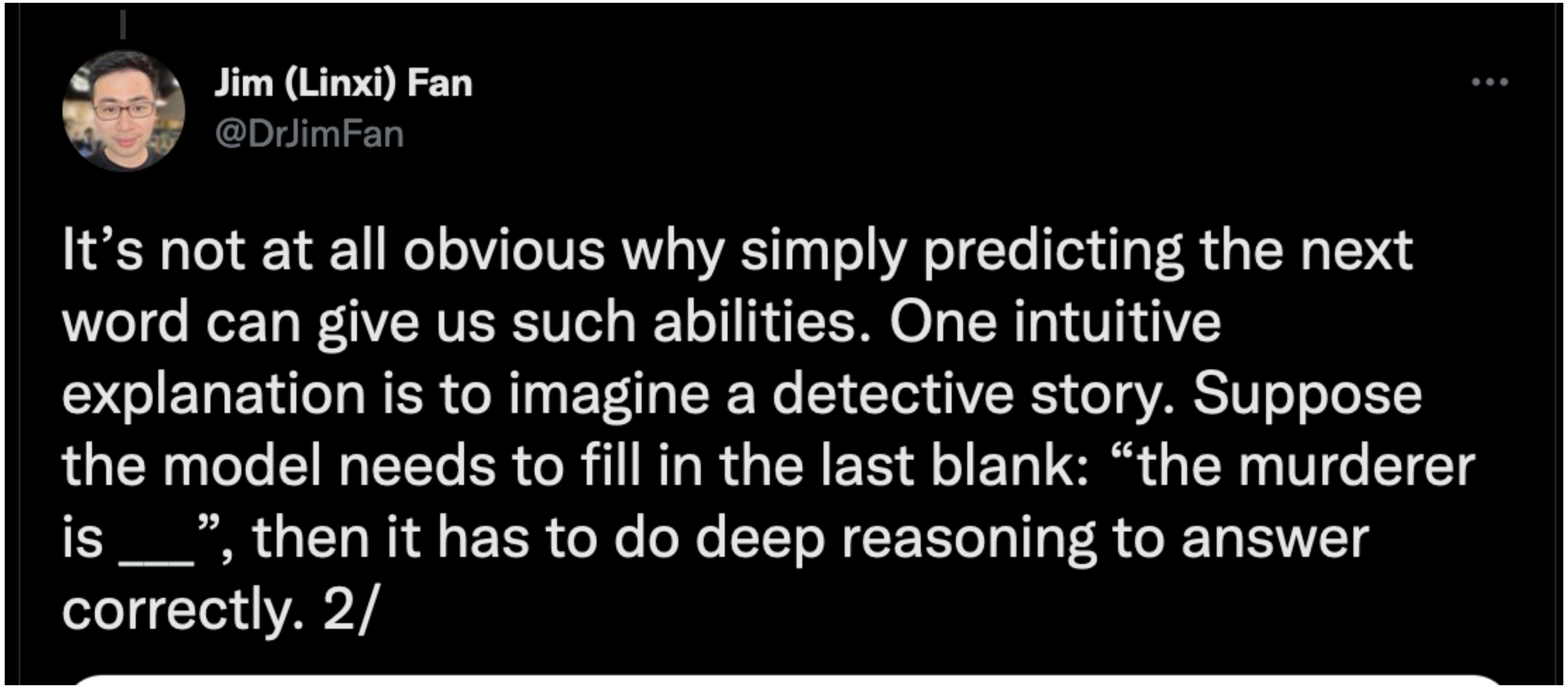
https://galactica.org/

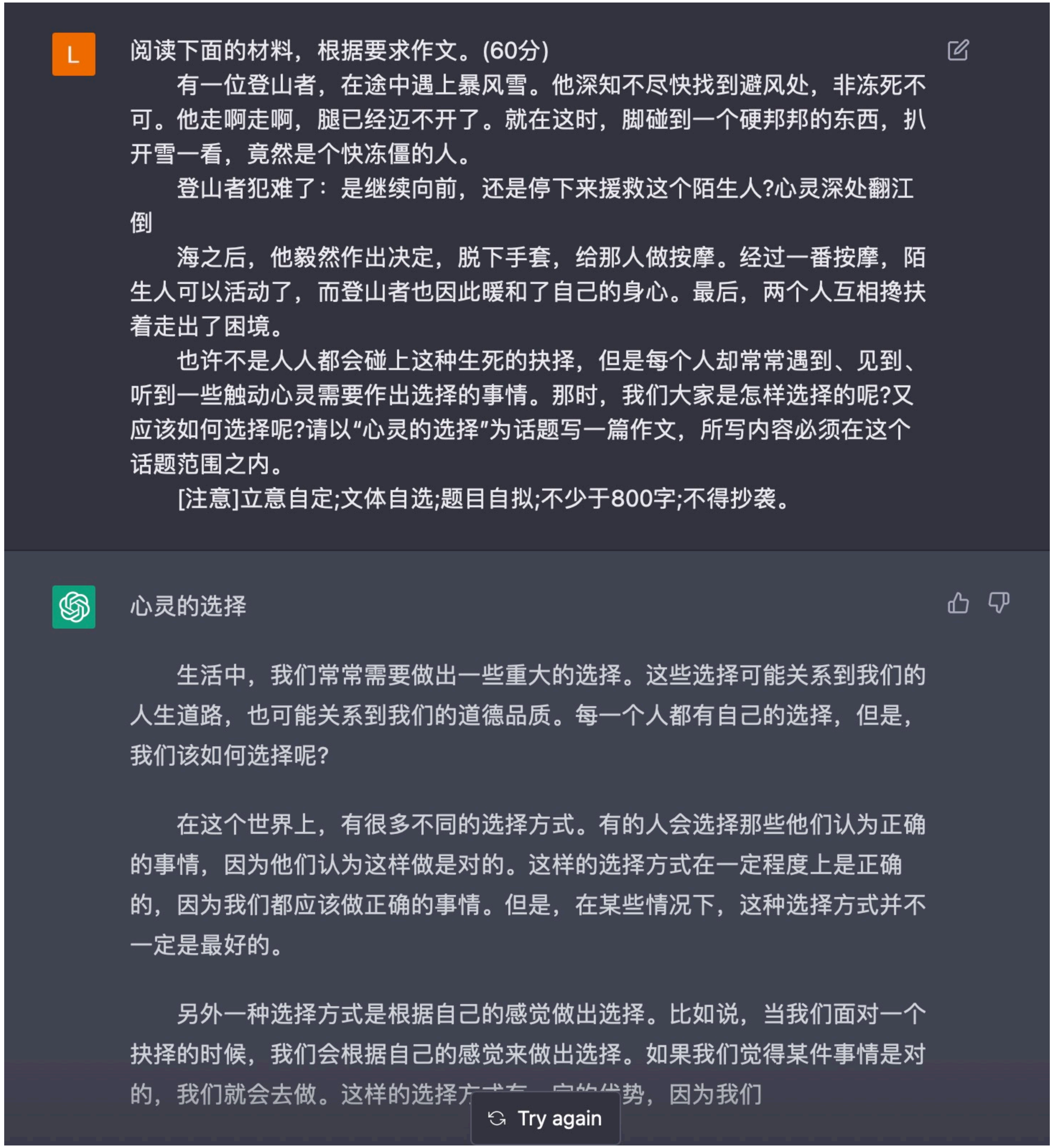
Galactica: A Large Language Model for Science

Meta, Nov 16, 2022

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







L

阅读下面的材料，根据要求作文。(60分)

有一位登山者，在途中遇上暴风雪。他深知不尽快找到避风处，非冻死不可。他走啊走啊，腿已经迈不开了。就在这时，脚碰到一个硬邦邦的东西，扒开雪一看，竟然是个快冻僵的人。

登山者犯难了：是继续向前，还是停下来援救这个陌生人?心灵深处翻江倒

海之后，他毅然作出决定，脱下手套，给那人做按摩。经过一番按摩，陌生人可以活动了，而登山者也因此暖和了自己的身心。最后，两个人互相搀扶着走出了困境。

也许不是人人都会碰上这种生死的抉择，但是每个人却常常遇到、见到、听到一些触动心灵需要作出选择的事情。那时，我们大家是怎样选择的呢?又应该如何选择呢?请以“心灵的选择”为话题写一篇作文，所写内容必须在这个话题范围之内。

[注意]立意自定;文体自选;题目自拟;不少于800字;不得抄袭。

心灵的选择

生活中，我们常常需要做出一些重大的选择。这些选择可能关系到我们的人生道路，也可能关系到我们的道德品质。每一个人都有自己的选择，但是，我们该如何选择呢?

在这个世界上，有很多不同的选择方式。有的人会选择那些他们认为正确的事情，因为他们认为这样做是对的。这样的选择方式在一定程度上是正确的，因为我们都应该做正确的事情。但是，在某些情况下，这种选择方式并不一定是最好的。

另外一种选择方式是根据自己的感觉做出选择。比如说，当我们面对一个抉择的时候，我们会根据自己的感觉来做出选择。如果我们觉得某件事情是对的，我们就会去做。这样的选择方[illegible]势，因为我们

Try again

prompt

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

https://chat.openai.com/chat

警察来到一个家庭凶杀案现场，发现五人出现过凶杀现场。

A说：D是凶手
B：不是我干的
C：不是 E 干的
D：A说我干的，纯属谎言
E：B说的是实话

警察确定，有三个人的陈述是事实
问：谁是凶手?

根据题目中给出的信息，我们可以推断出：

- 由于有三个人的陈述是事实，所以有且只有一个人是在说谎。
- 由于E说B说的是实话，所以E是在说实话。
- 由于D说A说的是谎话，所以A是在说谎。
- 由于A说D是凶手，所以D不是凶手。
- 由于D说A说的是谎话，所以A不是在说实话。

因此，A、B、C、D、E五个人中只有A和D是在说谎。而由于D说A说的是谎话，所以A不是在说实话。也就是说，A说的是谎话，即D不是凶手。因此，凶手不是D。

我们可以继续分析，如果凶手是B，那么B说的就是谎话，而E说的也是谎话。但是根据题目中给出的信息，E说的是实话，因此B不是凶手。

如果凶手是C，那么C说的就是实话，而E说的也是实话。但是根据题目中给出的信息，E说的是实话，因此C不是凶手。

如果凶手是E，那么E说的就是谎话，而B说的也是谎话。但是根据题目中给出的信息，B说的是实话，因此E不是凶手。

综上所述，A、B、C、D、E五个人中只有A和D是在说谎，B、C、E三个人的陈述是事实。
因

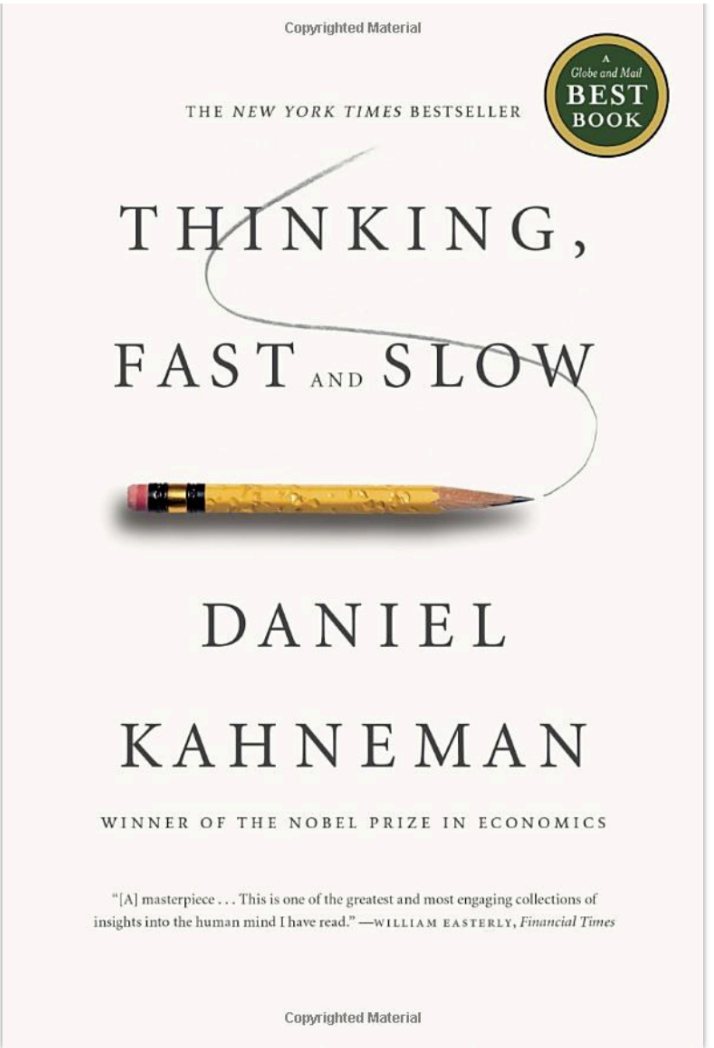
继续

此，凶手是A、B、C、D、E五个人中的一个人，且不是D。根据题目中给出的条件，我们无法确定凶手到底是谁。因此，我们无法确定谁是凶手。

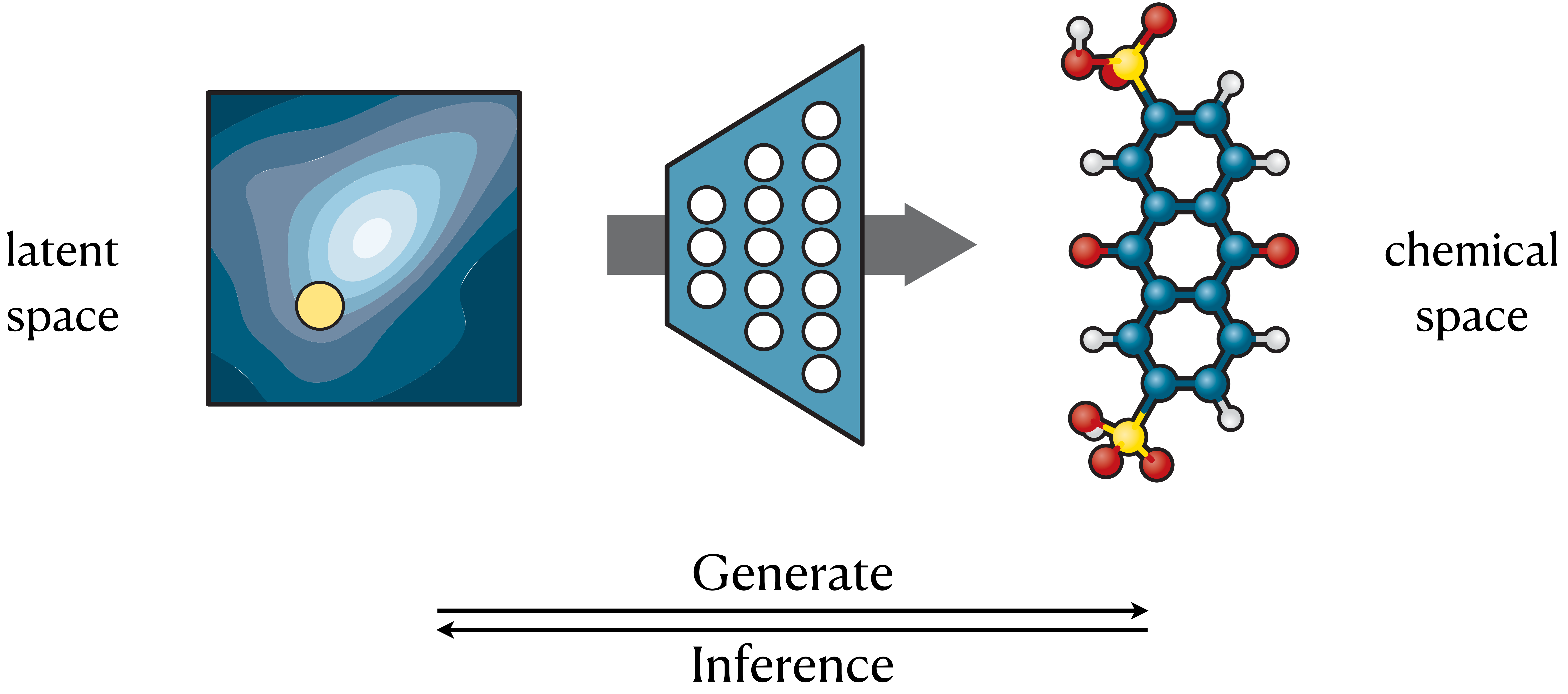
BTW, a failed example of chatGPT

Is it possible to make it work with prompt engineering ?

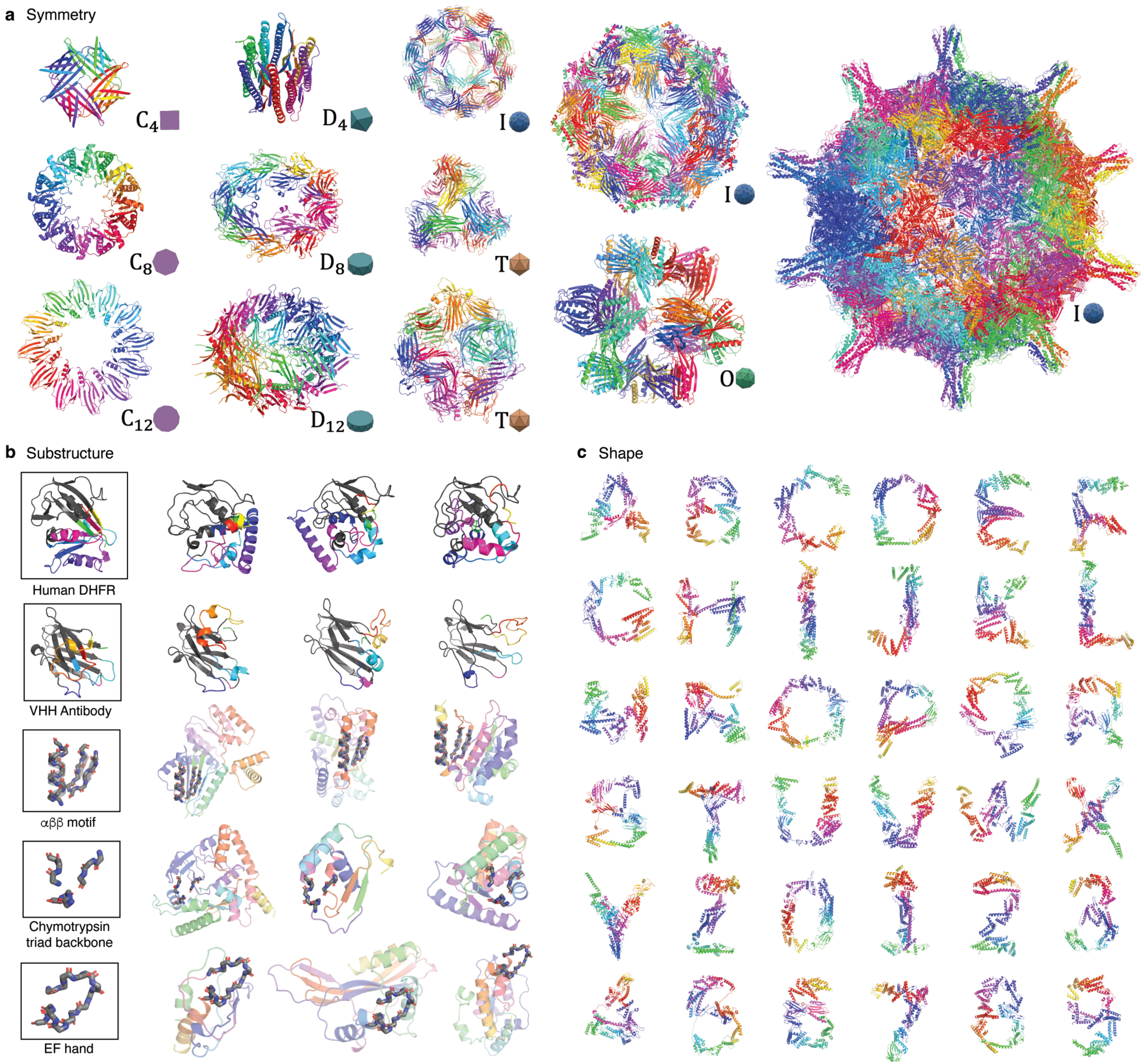
How to integrate symbolic logic into large language models ?



Generative AI for matter engineering



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

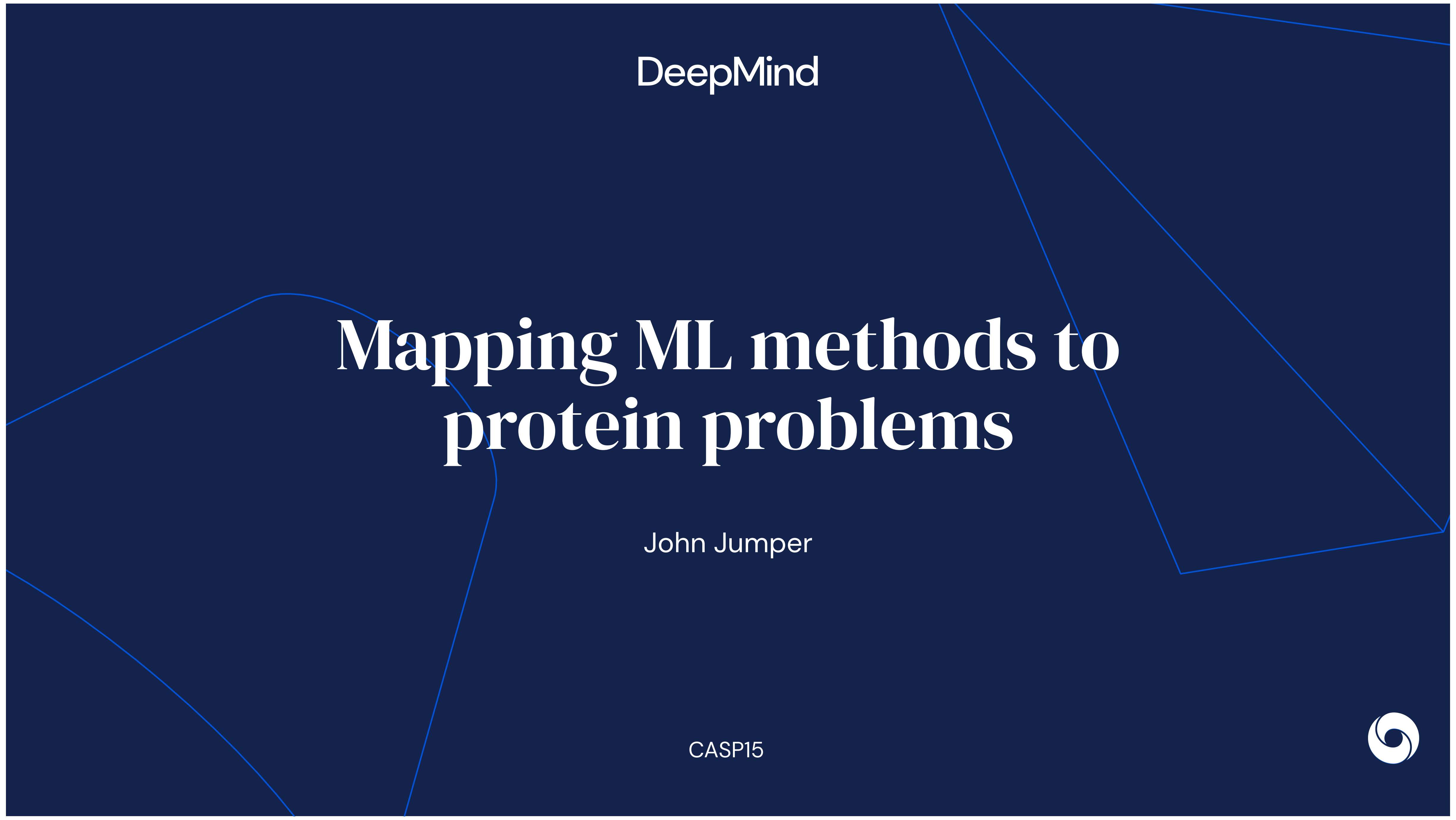


$p(\text{protein} \mid \text{symmetry})$

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

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

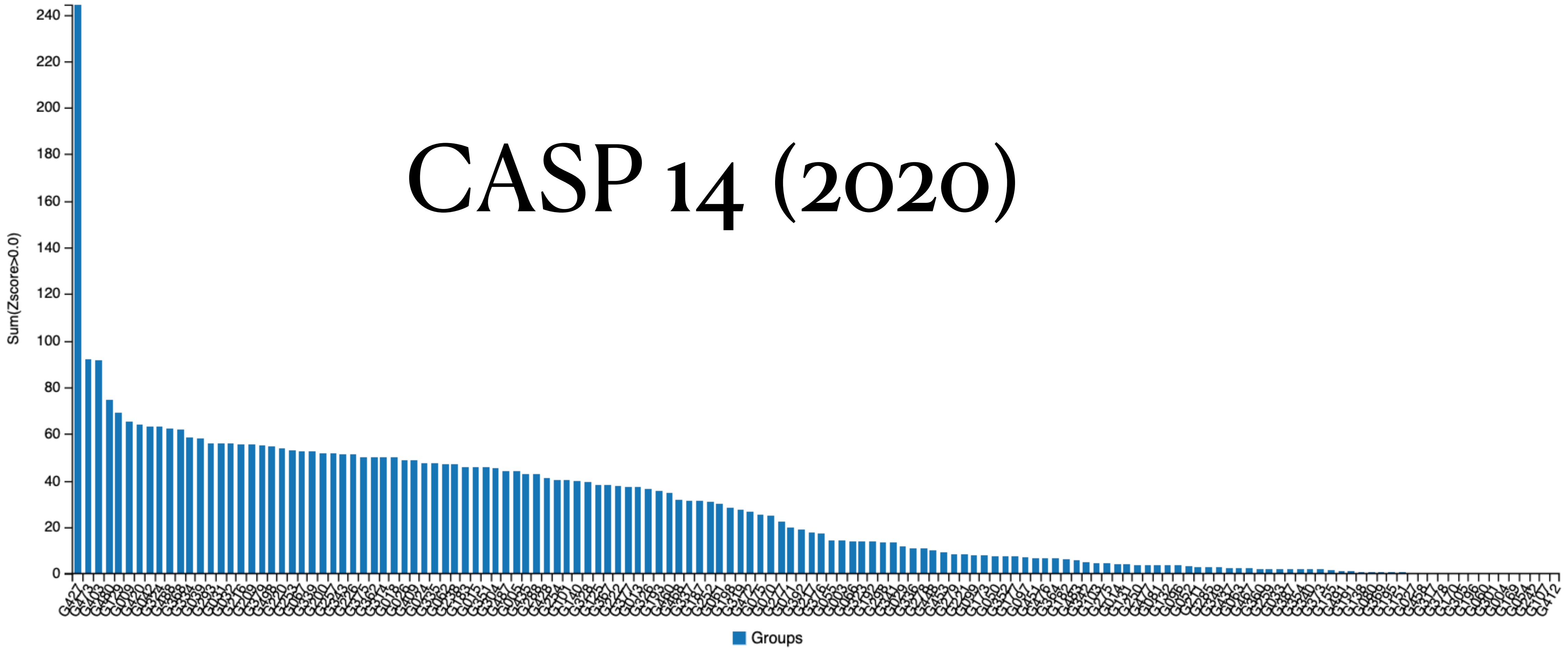
https://generatebiomedicines.com/chroma

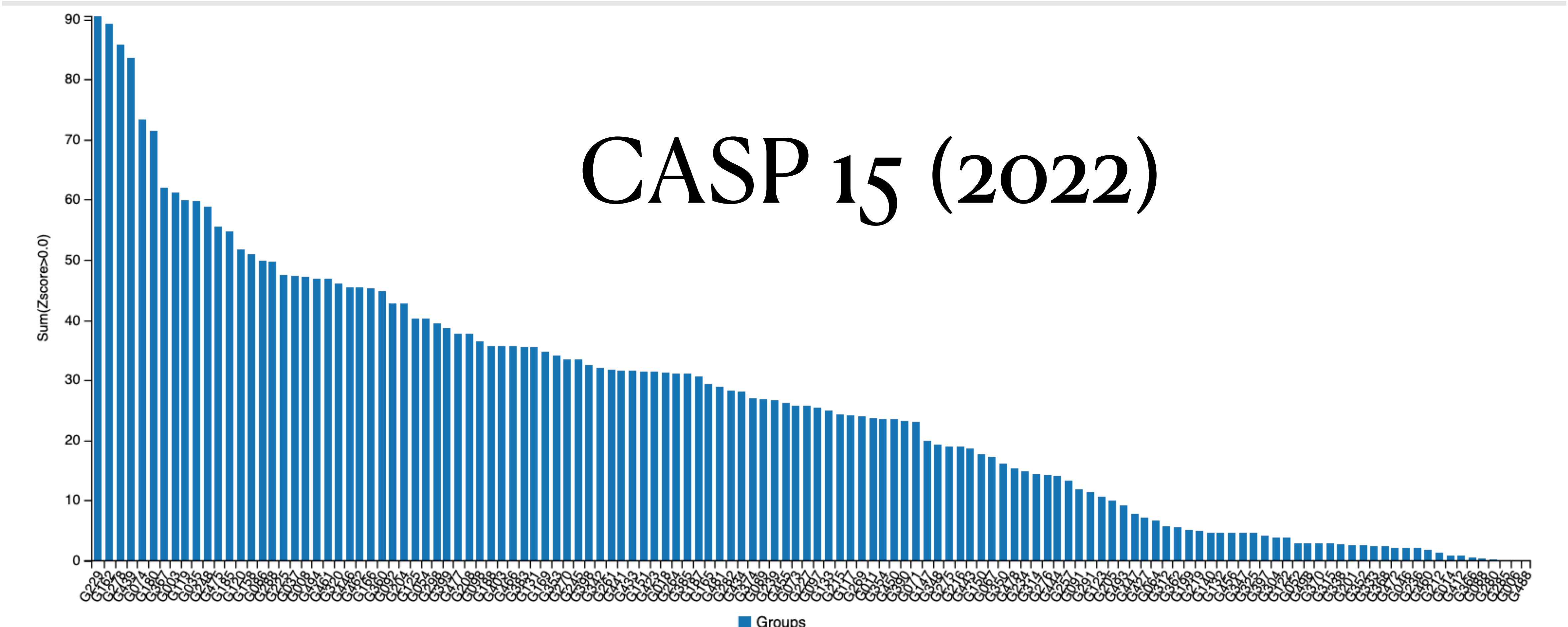


CASP 15 invited talk by John Jumper

Outline

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





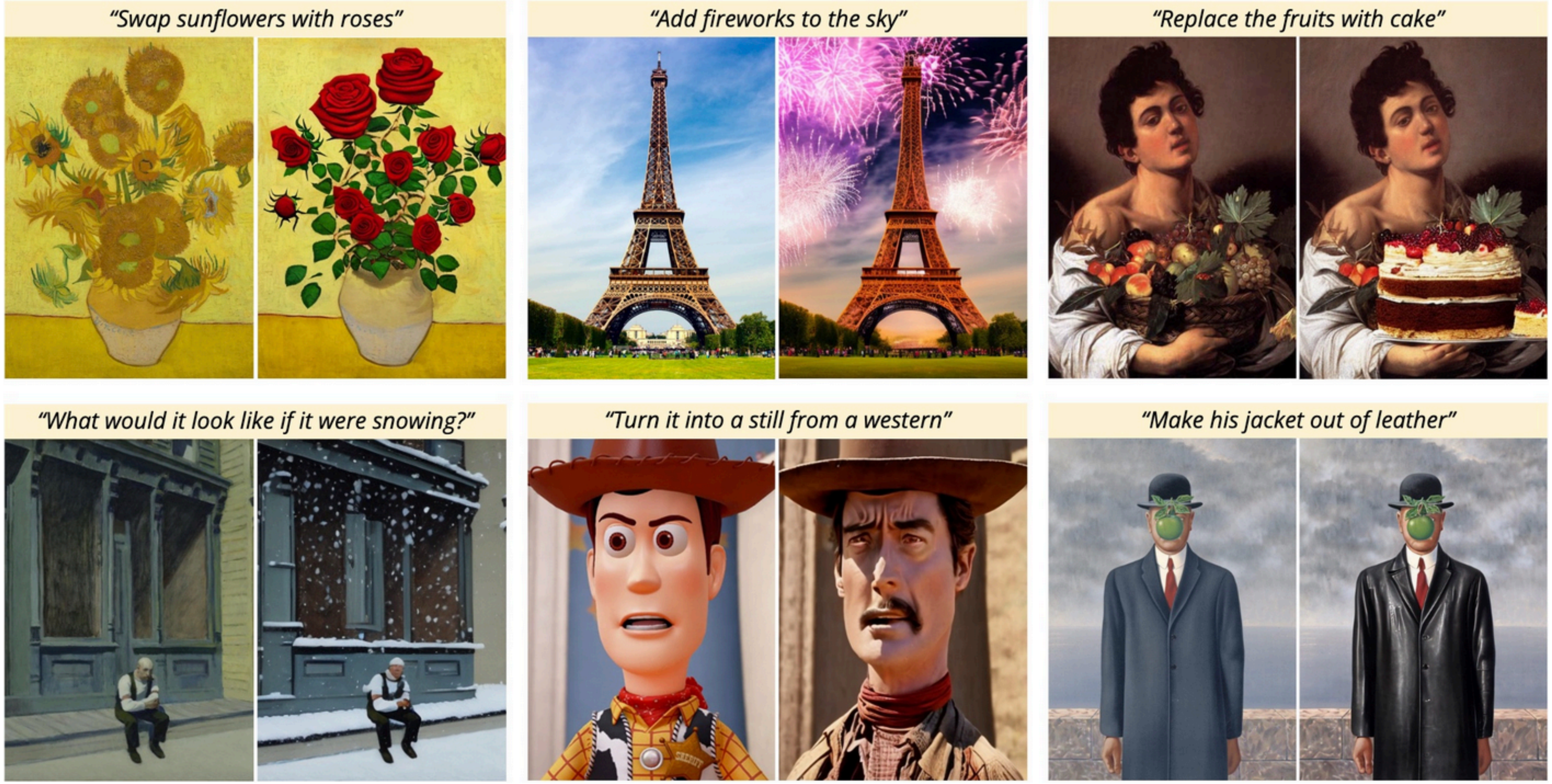
Great to have for protein generative models

textual inversion

https://textual-inversion.github.io/

Great to have for protein generative models

Instruct-pix2pix

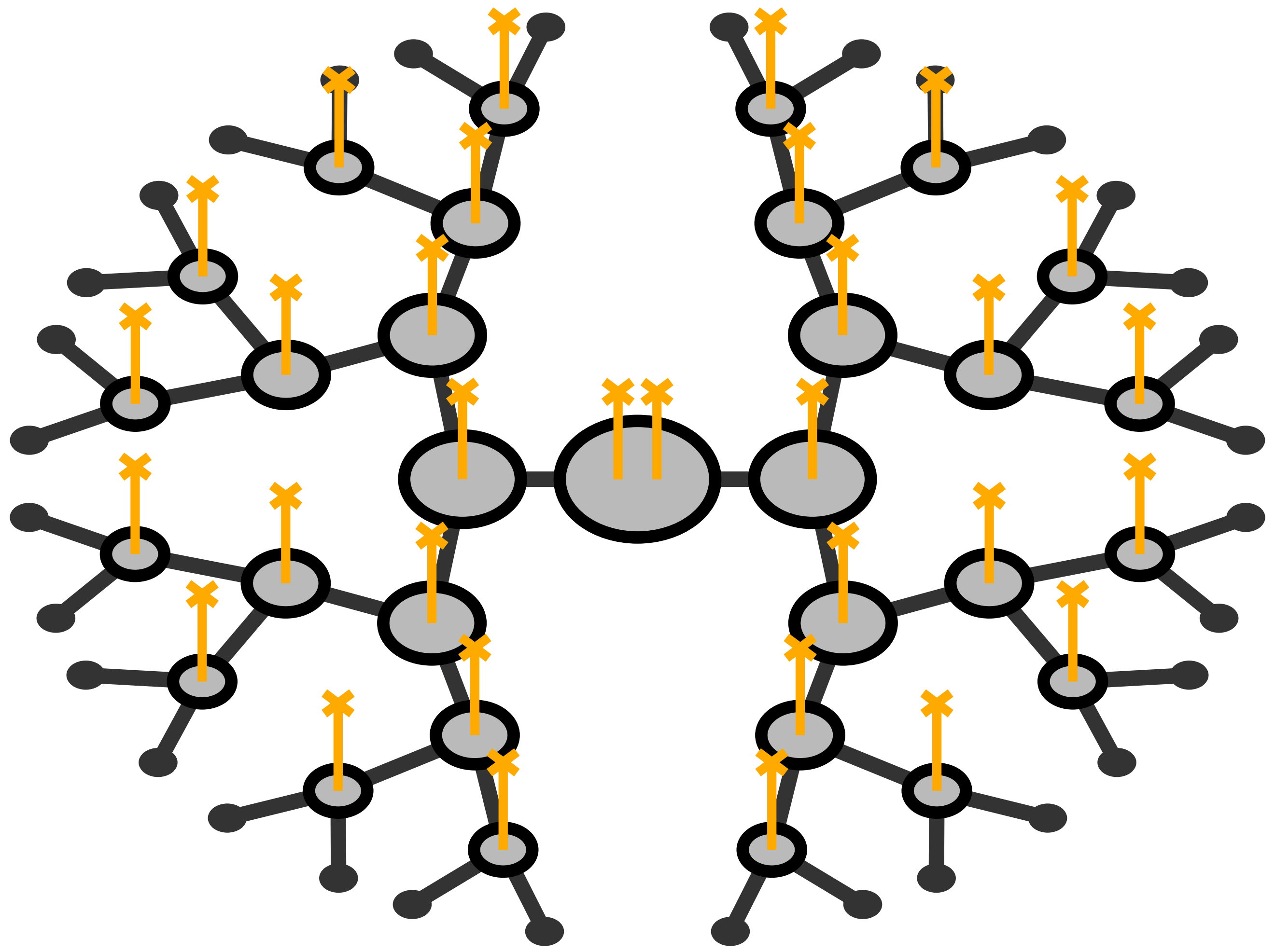


Given an image and a written instruction, our method follows the instruction to edit the image.

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

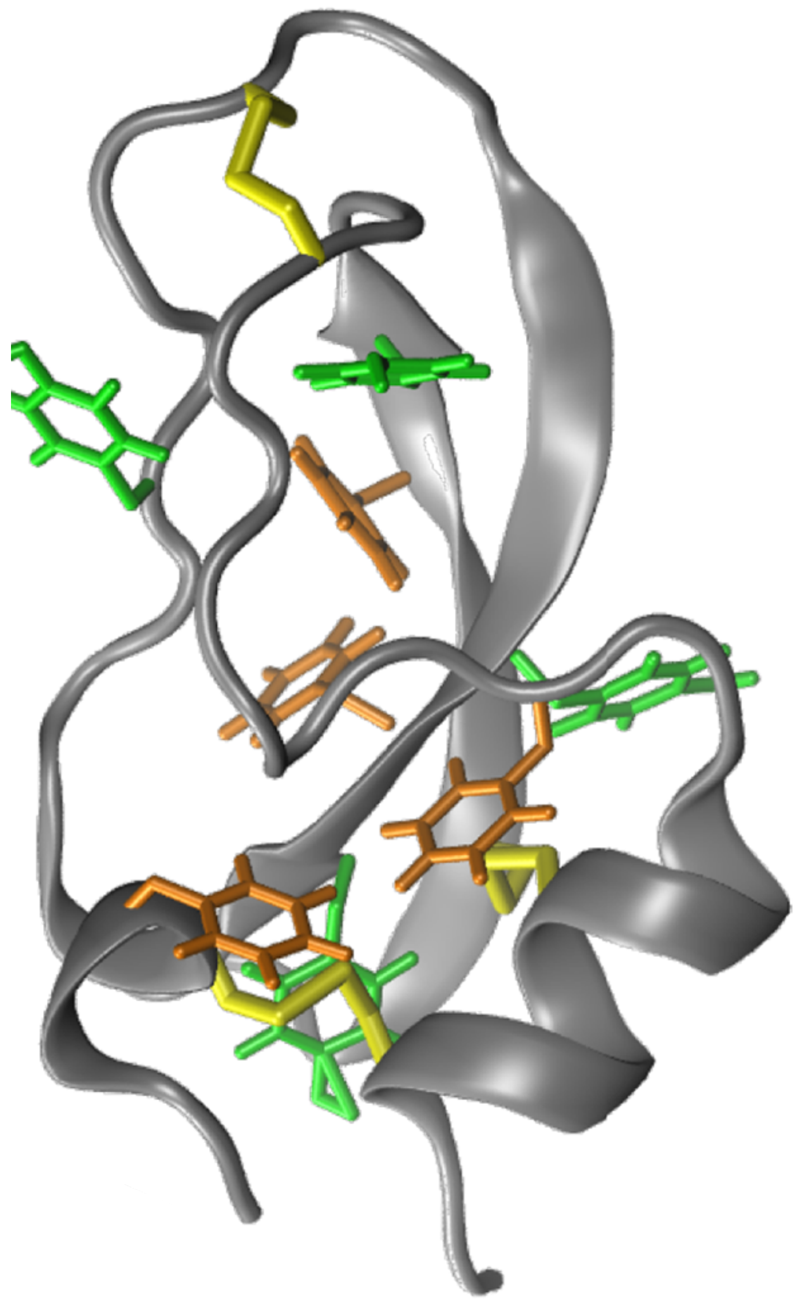
Generative AI for matter computation

Renormalization group



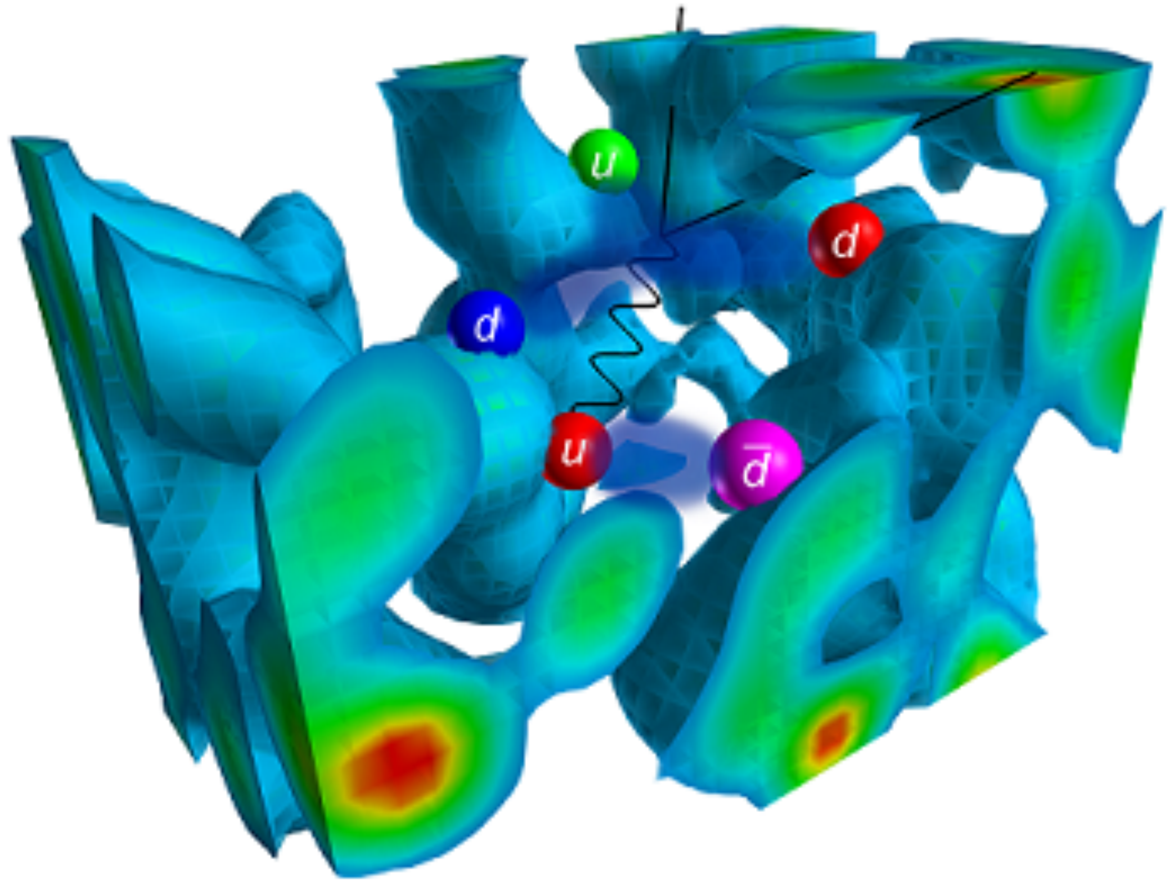
Li and LW, PRL '18
Li, Dong, Zhang, LW, PRX '20

Molecular simulation



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

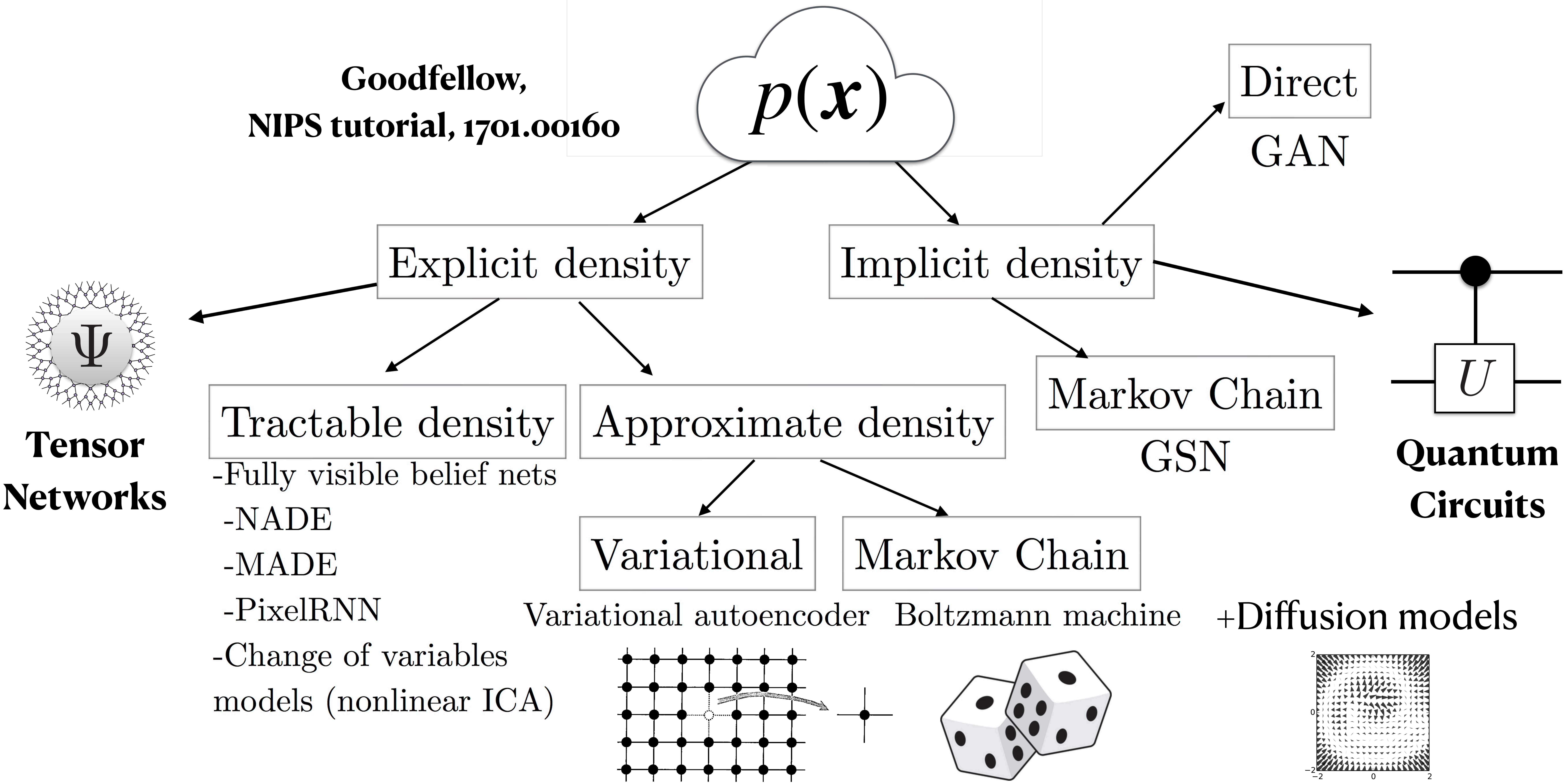
Lattice field theory



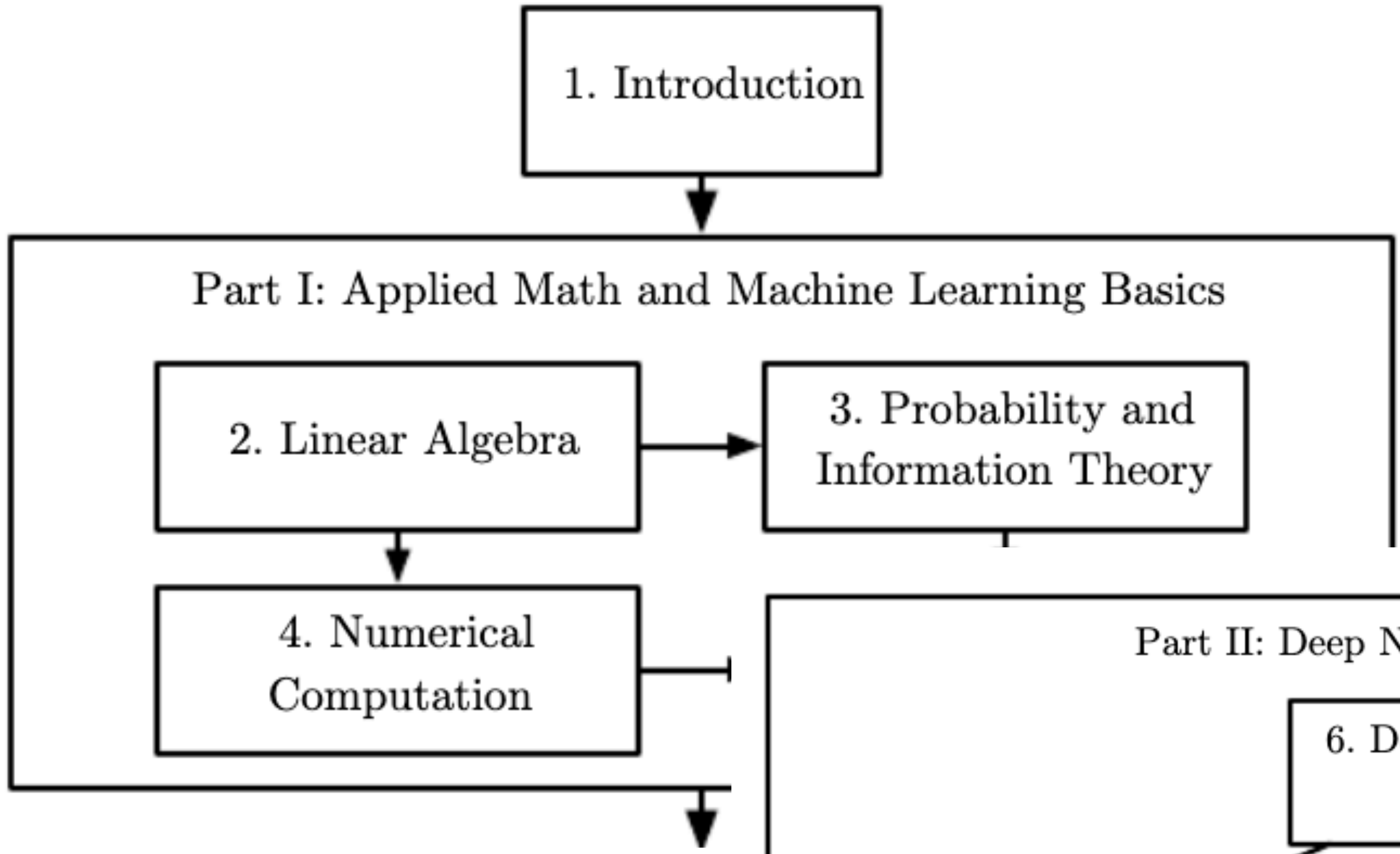
Albergo et al, PRD '19
Kanwar et al, PRL '20

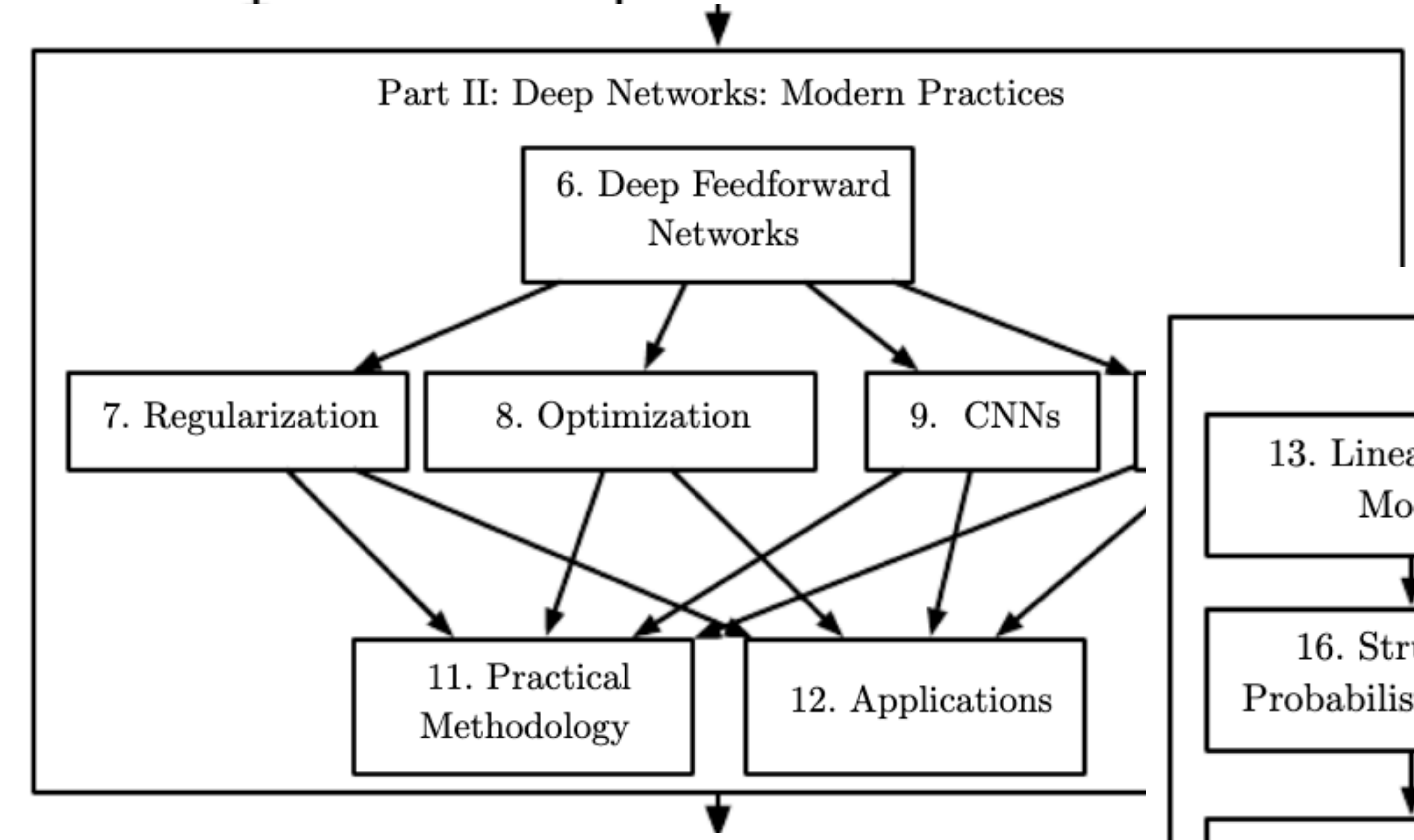
These are principled computation: quantitatively accurate, interpretable, reliable, and generalizable even without data

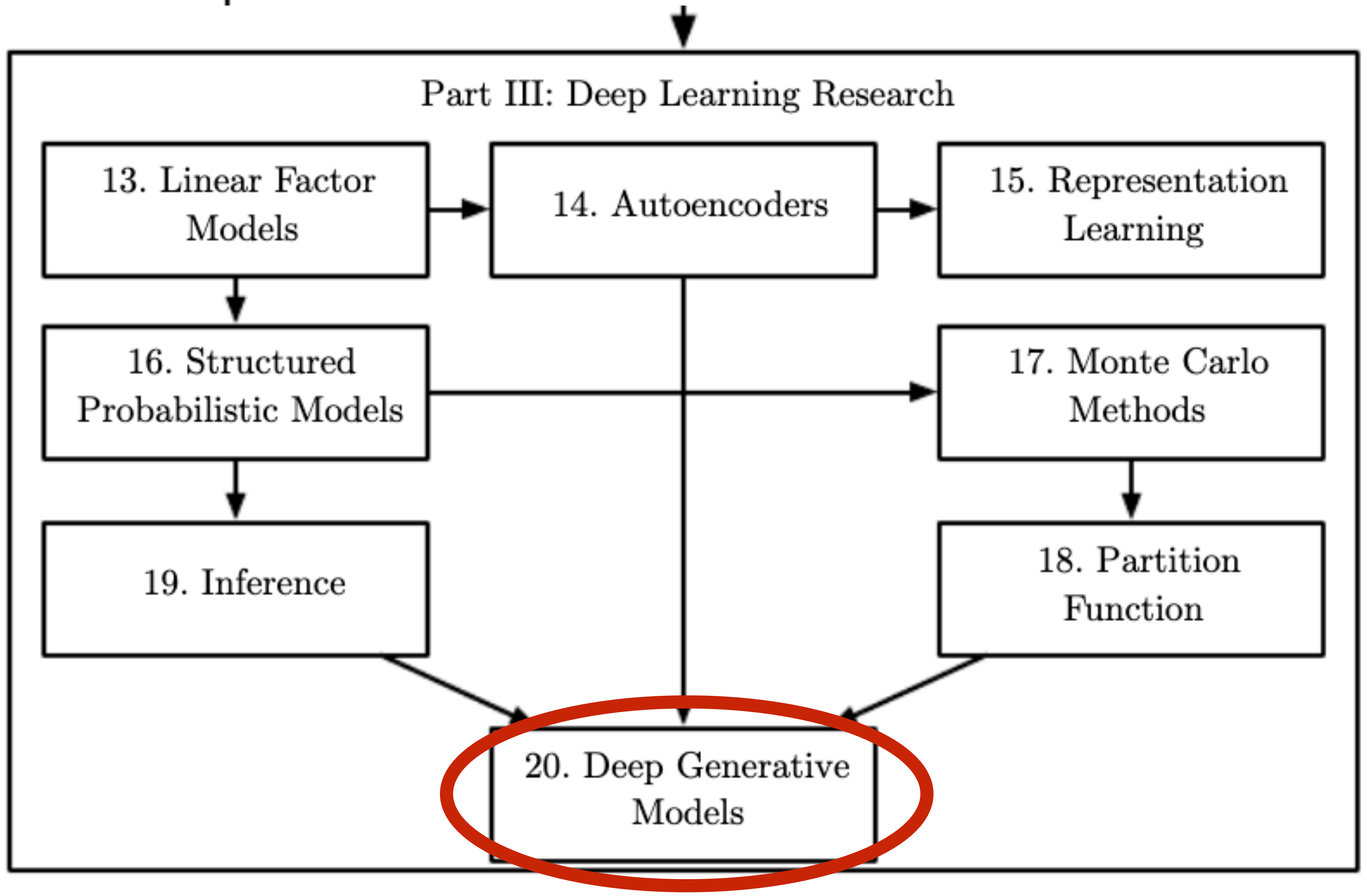
Generative models and their physics genes

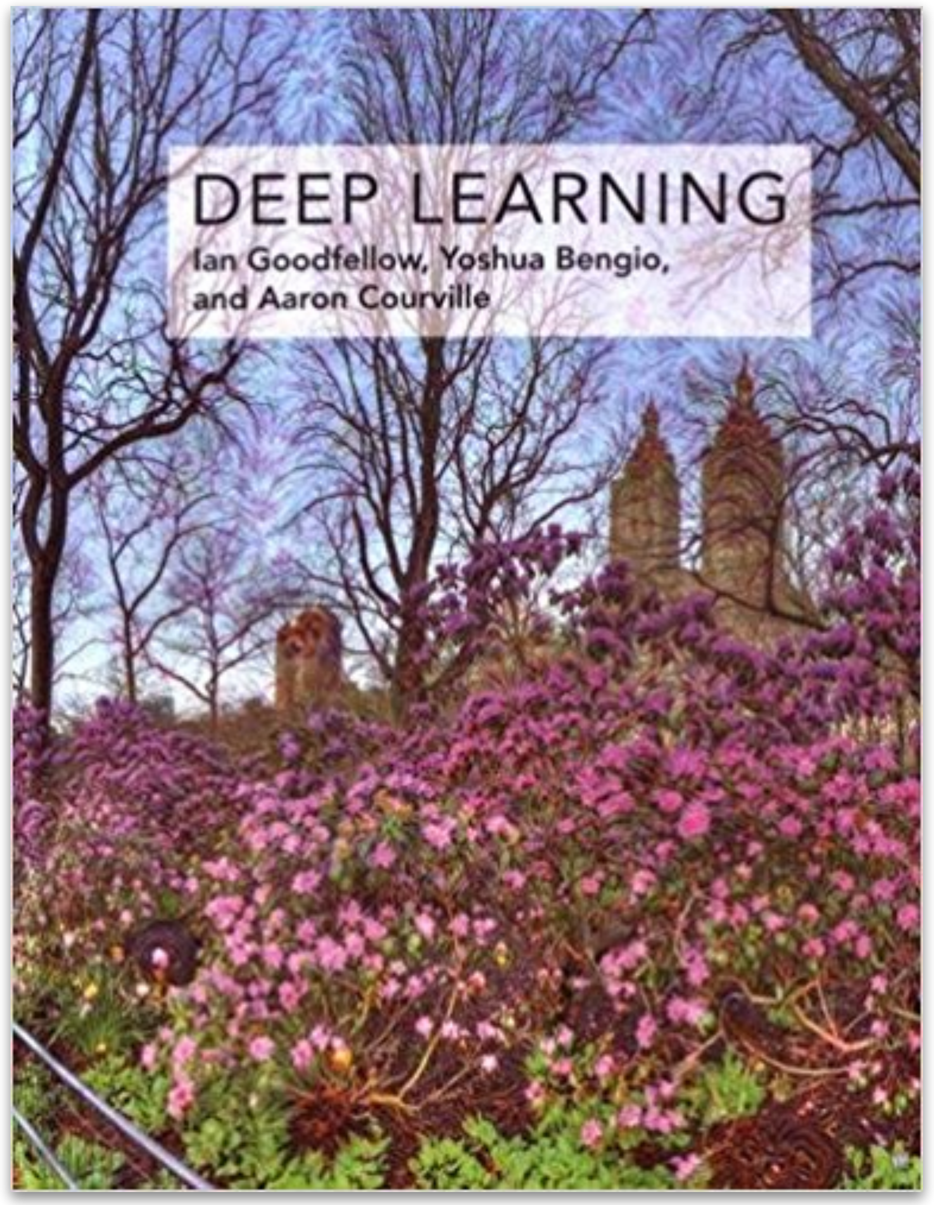


“Part III is the most important for a researcher—someone who wants to understand the breadth of perspectives that have been brought to the field of deep learning, and push the field forward towards true artificial intelligence.”









(outdated*) 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.

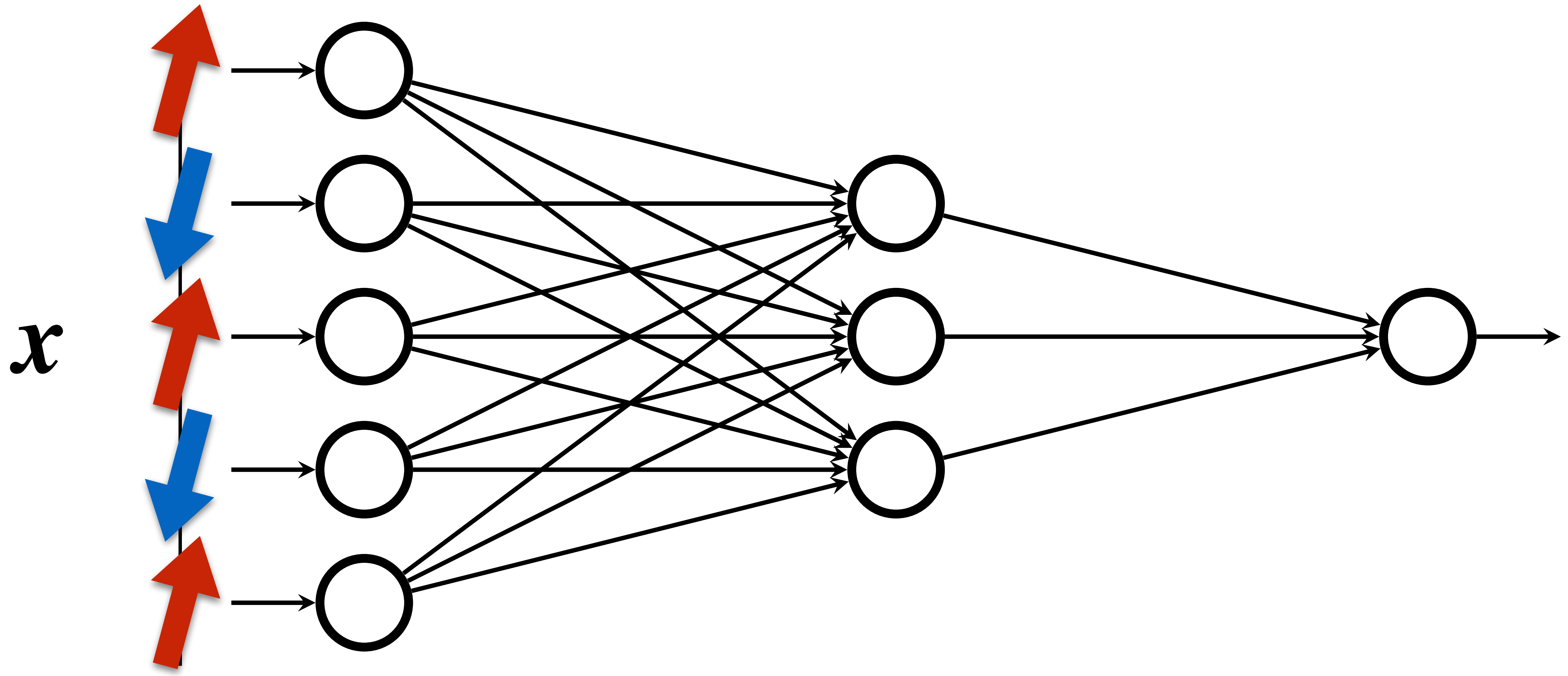
This note introduces the concept and principles of generative modeling, together with applications of modern generative models (autoregressive models, normalizing flows, variational autoencoders etc) as well as the old ones (Boltzmann machines) to physics problems. As a bonus, this note puts some emphasize 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

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

So, what is the fuss ?



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

Normalization ?

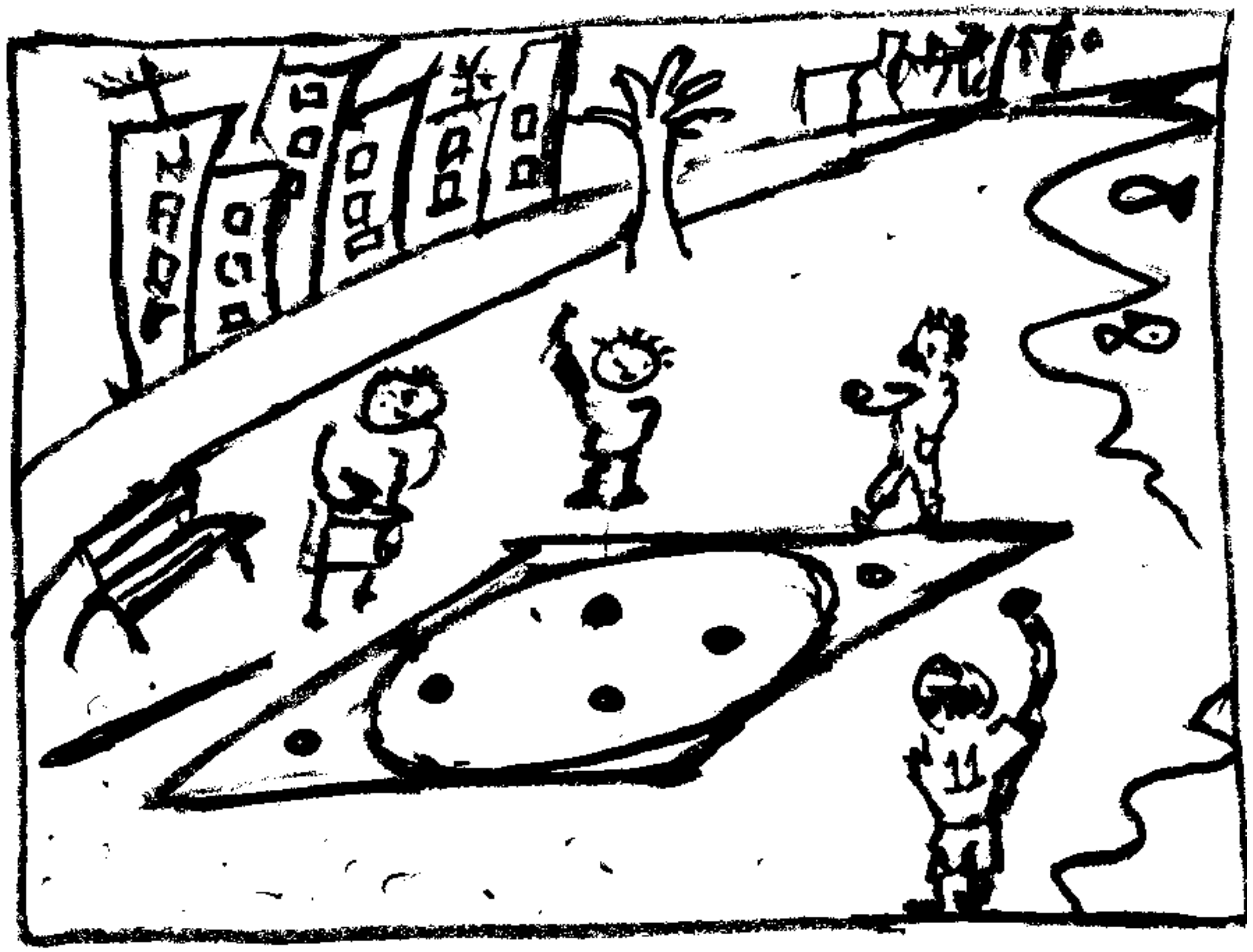
$$\int d\boldsymbol{x}\, p(\boldsymbol{x}) = 1$$

Sampling ?

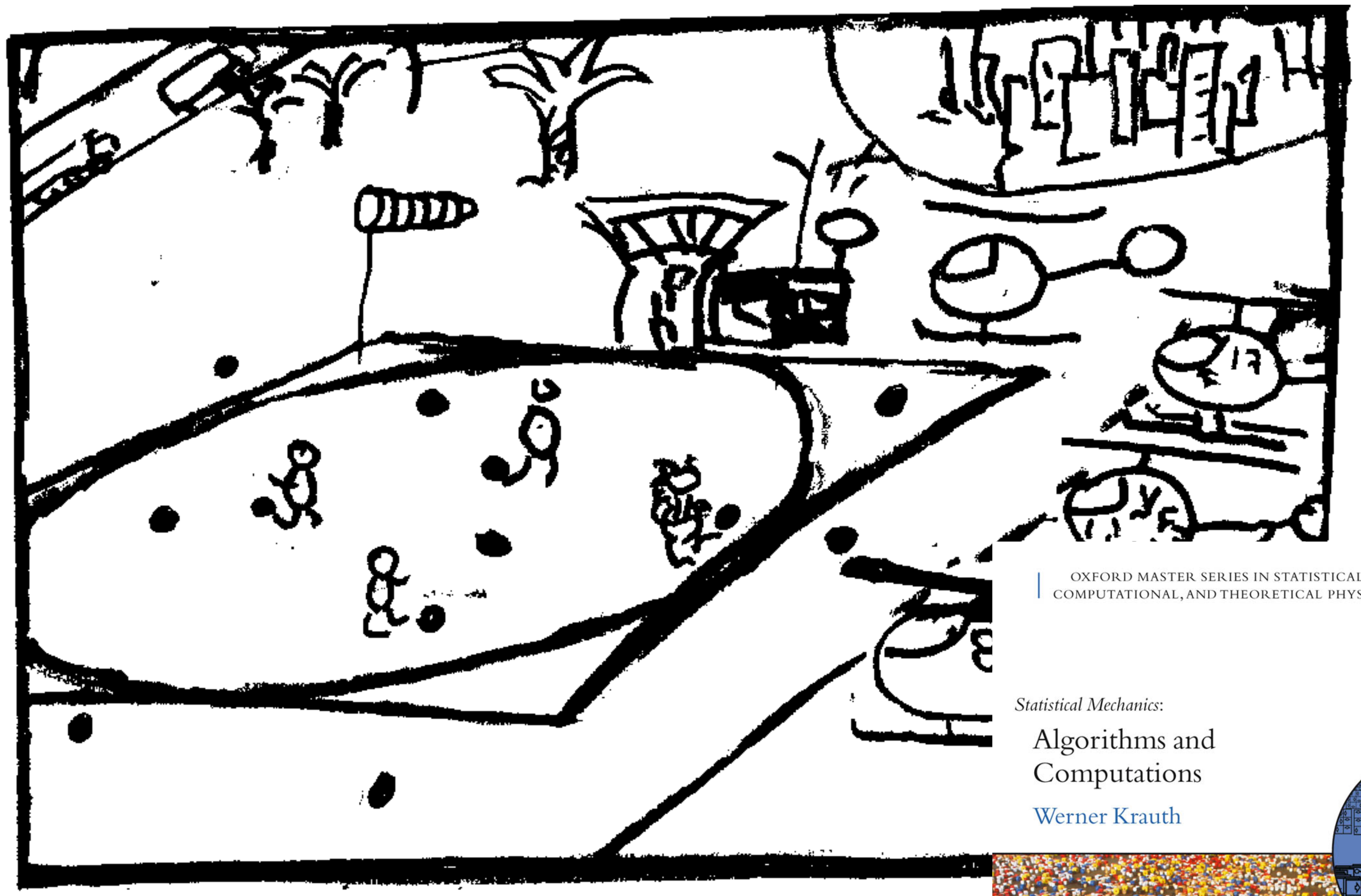
$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p(\boldsymbol{x})}$$

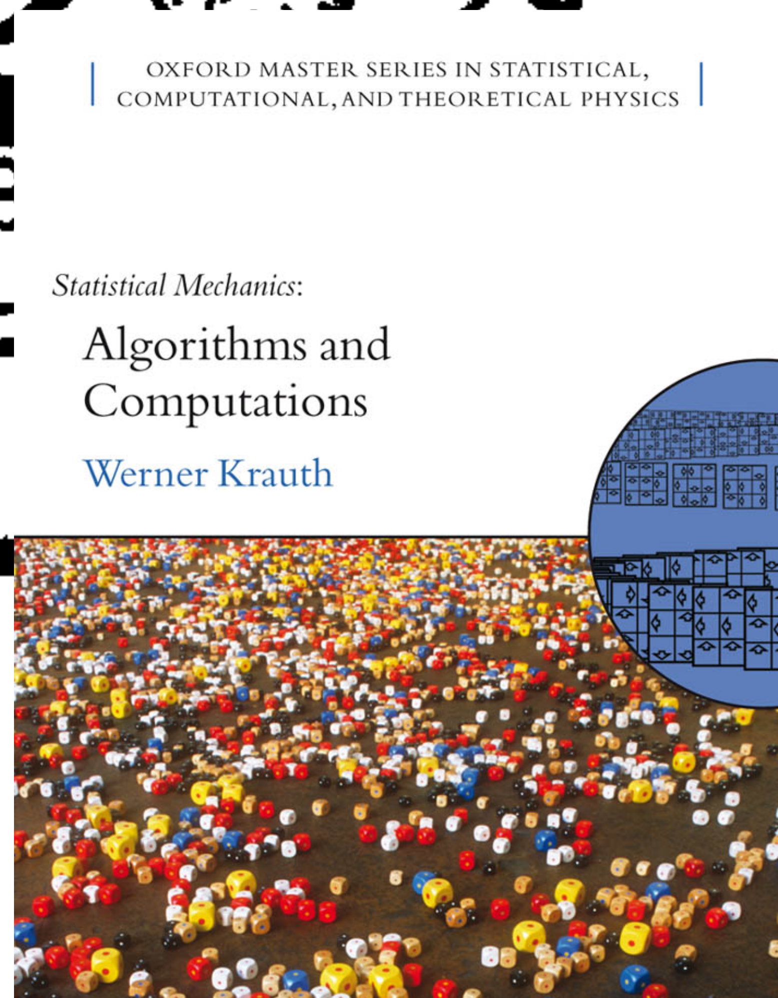
Perfect versus imperfect sampling

Children computing the number π on the Monte Carlo beach.



Adults computing the number π at the Monte Carlo heliport.





Generative modeling	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

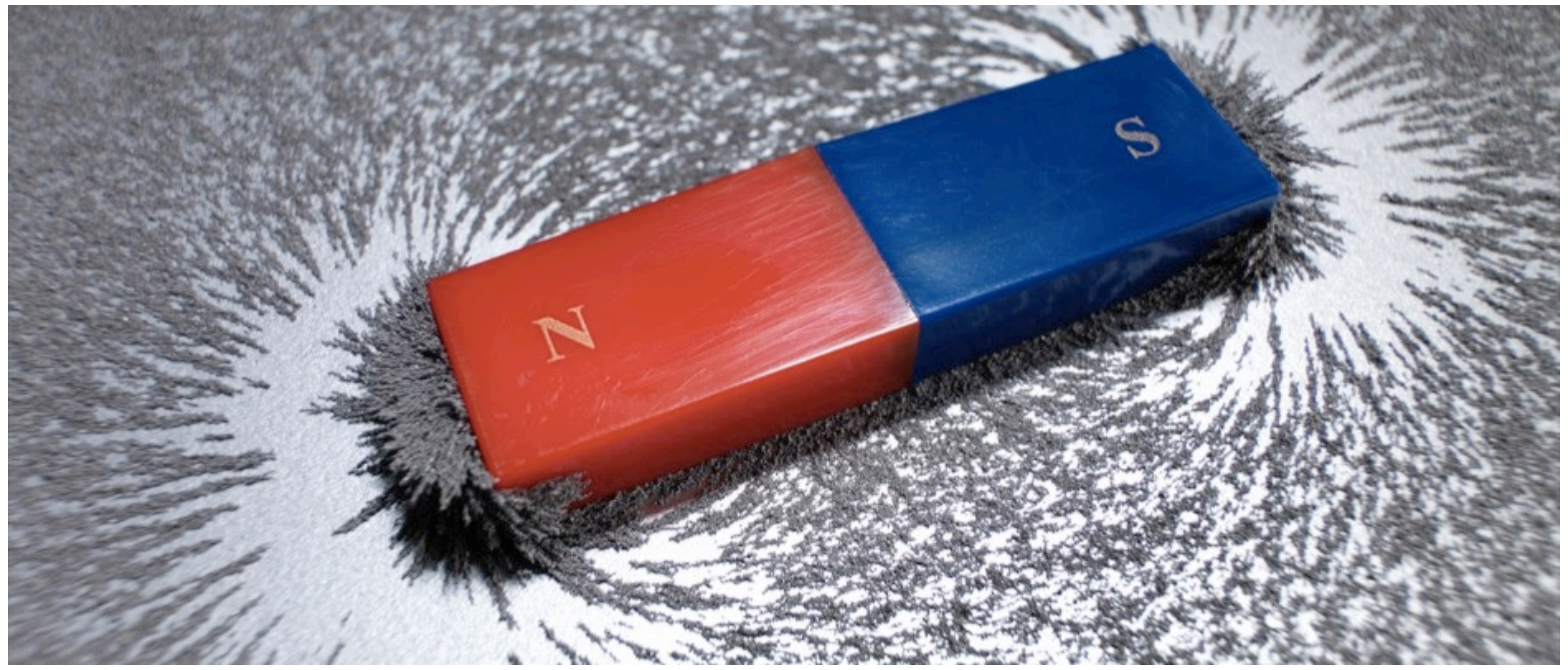
Known: samples

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 Hamiltonian"

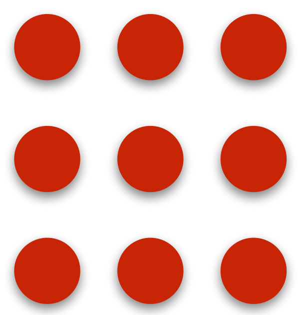
$$F = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[H(\boldsymbol{x}) + k_B T \ln p(\boldsymbol{x})\right]$$

$$\mathbb{KL}(\text{data} \parallel p) \quad \text{vs} \quad \mathbb{KL}(p \parallel e^{-H/k_BT})$$

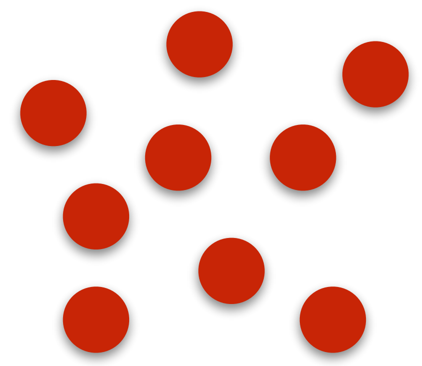
Nature tries to minimize free energy

$$F = E - TS$$

energy



entropy



F is a cost function given by Nature

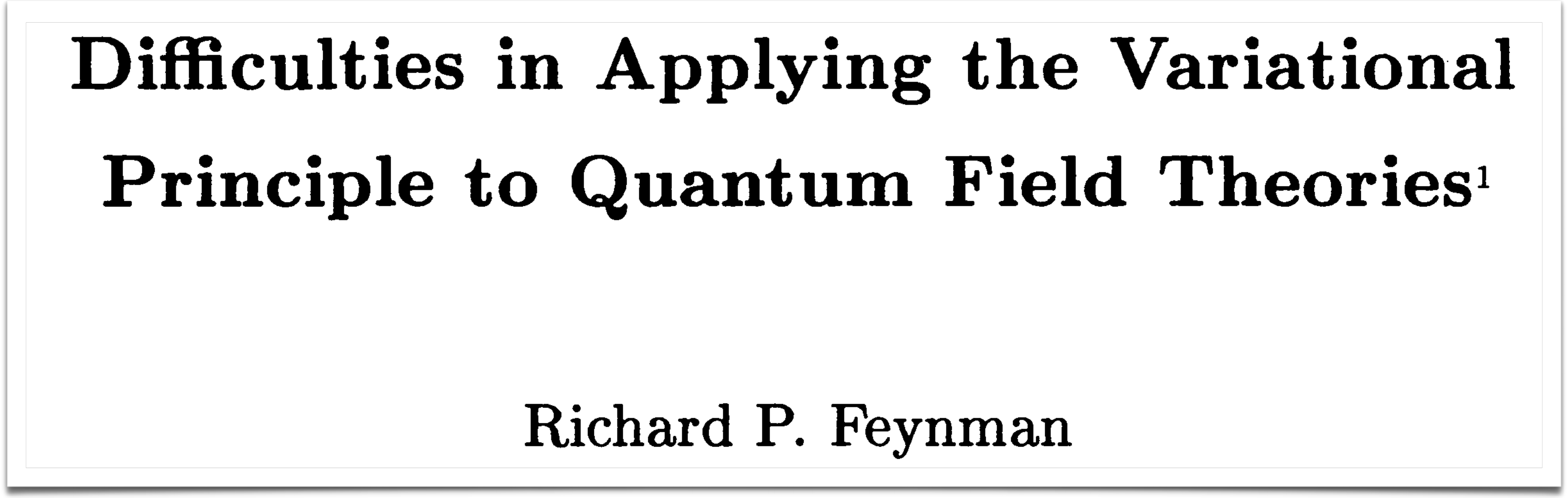
The variational free energy principle

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

↓ variational density

↓ energy

↓ entropy 😱


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

Richard P. Feynman

[1]transcript of his talk in 1987

Generative models!

Deep variational free energy approach

Use deep generative models as the variational density

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

$H(\boldsymbol{x})$ ↓ energy

$k_B T \ln p(\boldsymbol{x})$ ↓ entropy 😁

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

with normalizing flow & autoregressive models

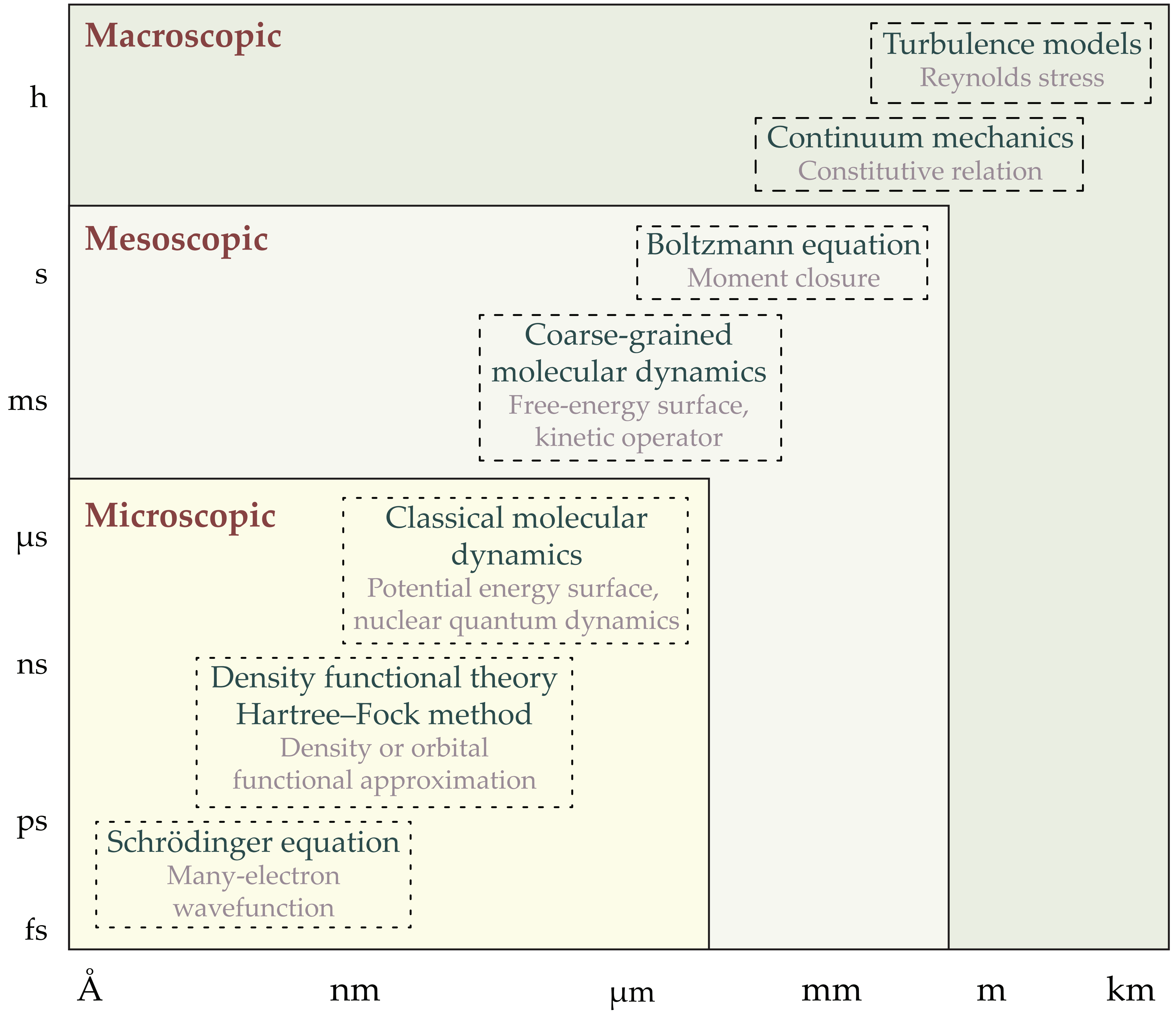
✓ Tractable entropy

✓ Direct sampling

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

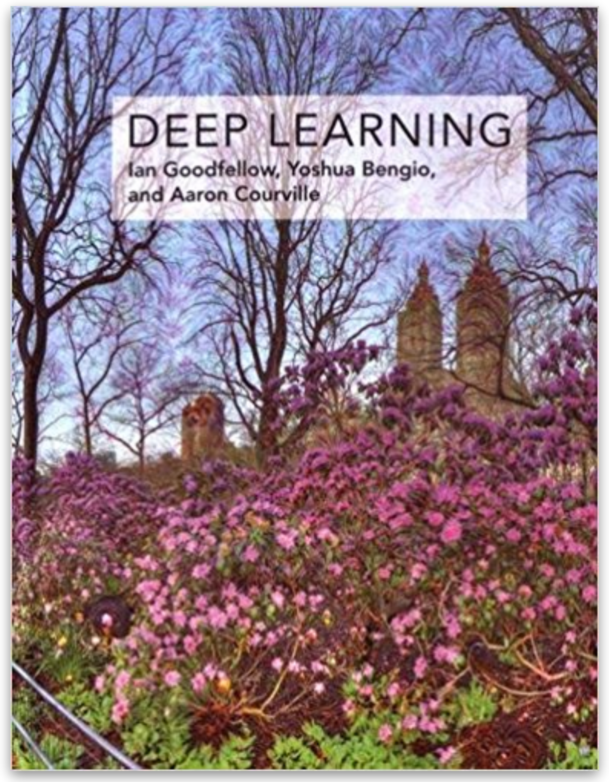
Deep variational free-energy in the context

E, Han,Zhang, Physics Today 2020



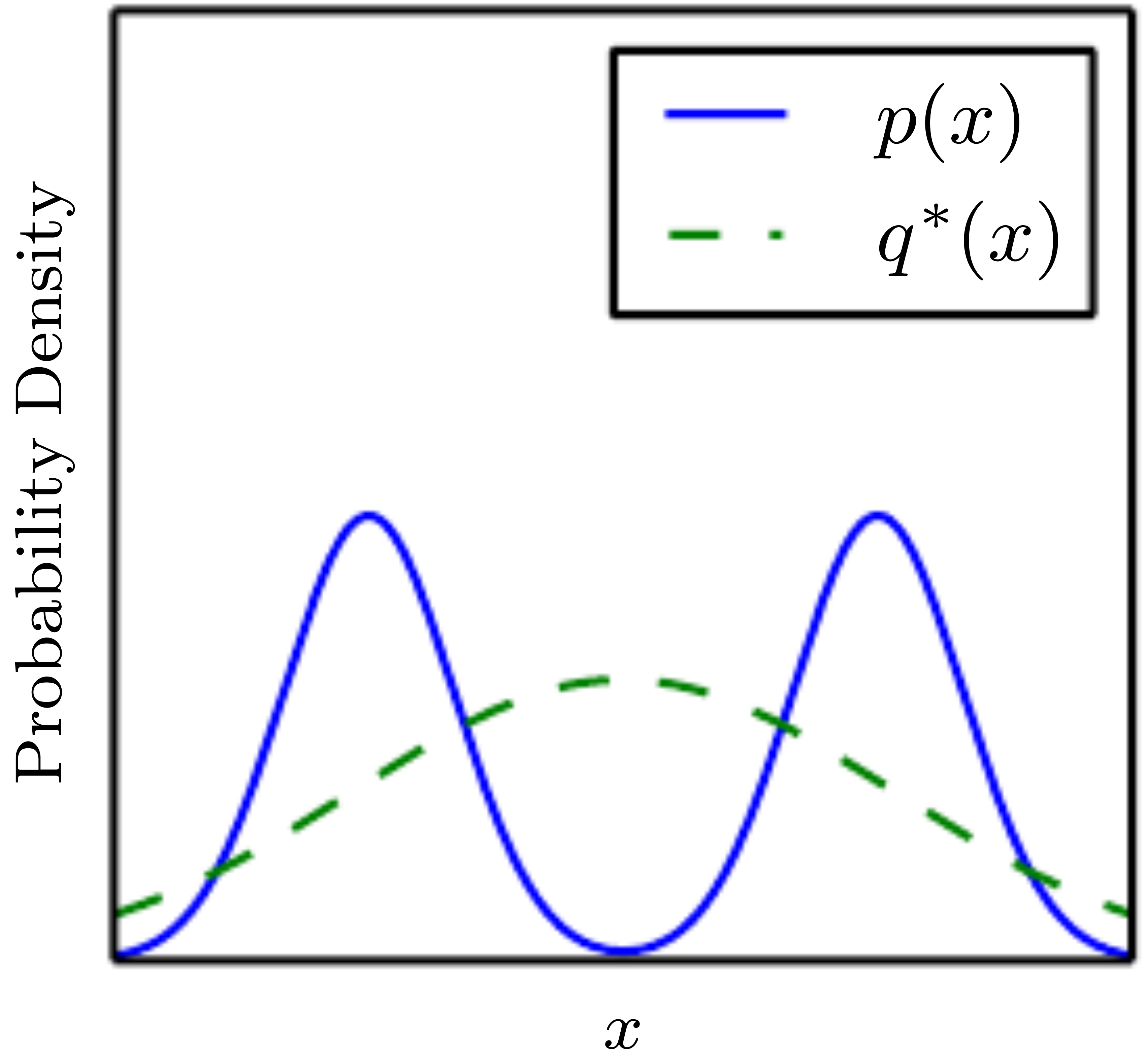
Objective	Model	Data	Task
MD potential energy surface	3N-dim function	DFT energy/ force	Generalization
DFT xc energy functional	3-dim functional	QMC/ CCSD/...	Generalization
Variational free-energy	3N-dim functional	No	Optimization

Forward KL or Reverse KL ?



Maximum likelihood estimation

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



Variational free energy

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

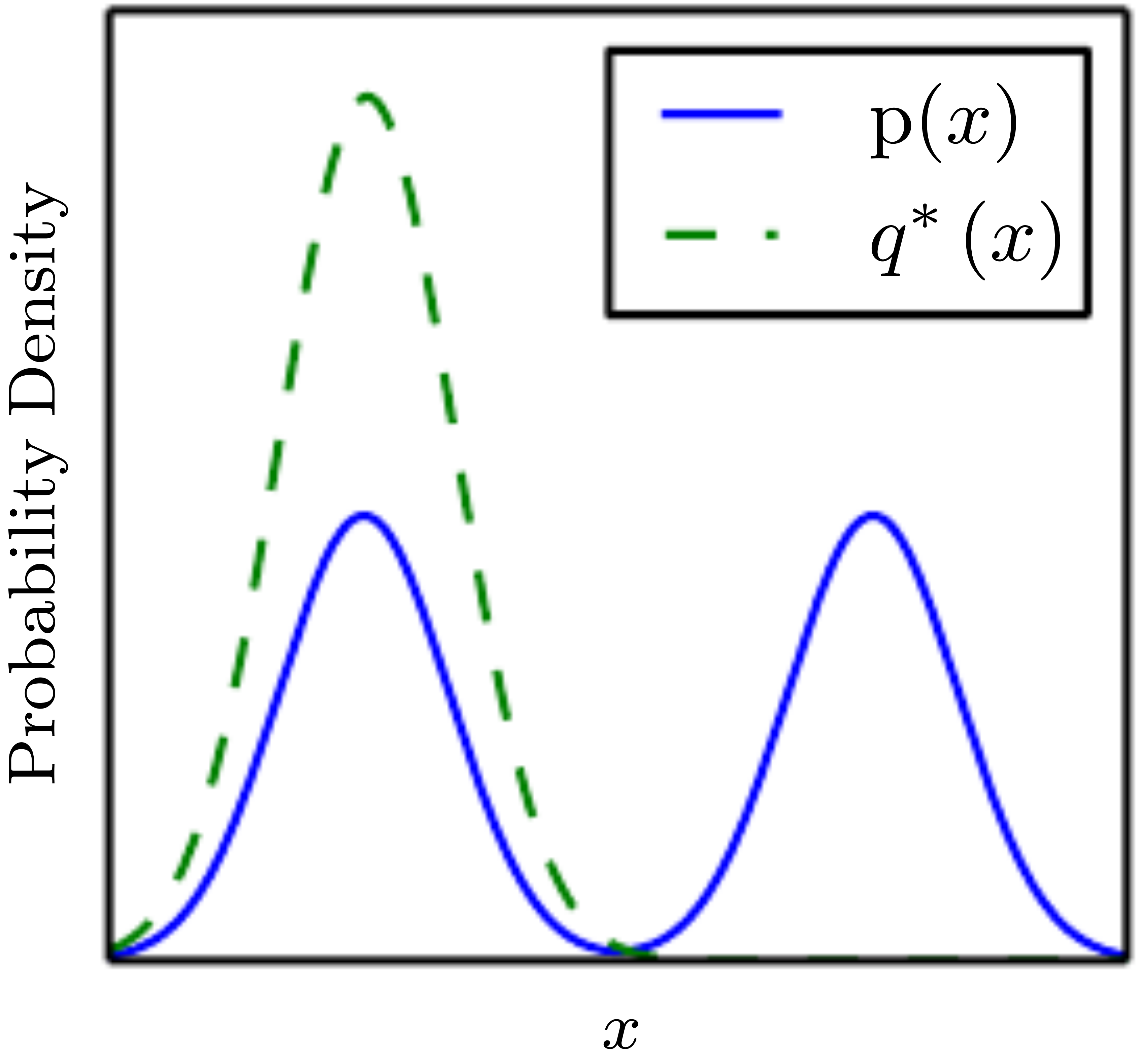


Fig. 3.6, Goodfellow, Bengio, Courville, http://www.deeplearningbook.org/

α-divergence

Minka, Microsoft Research Technical Report 2005

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

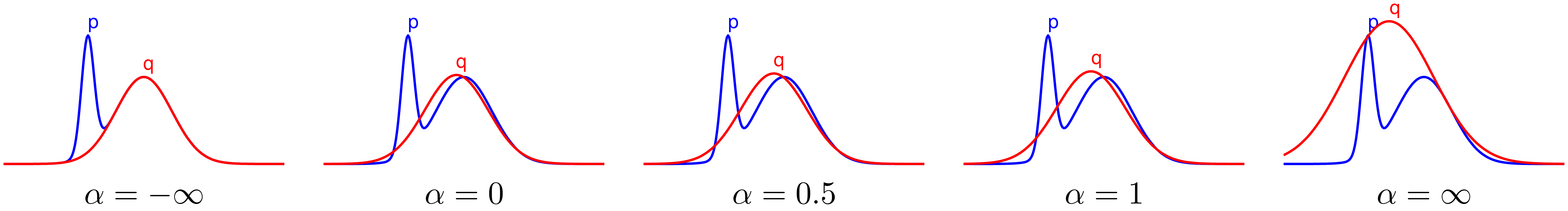
$$D_{-1}(p\,||\,q) = \frac{1}{2}\int_x \frac{(q(x)-p(x))^2}{p(x)} dx$$

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

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

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

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



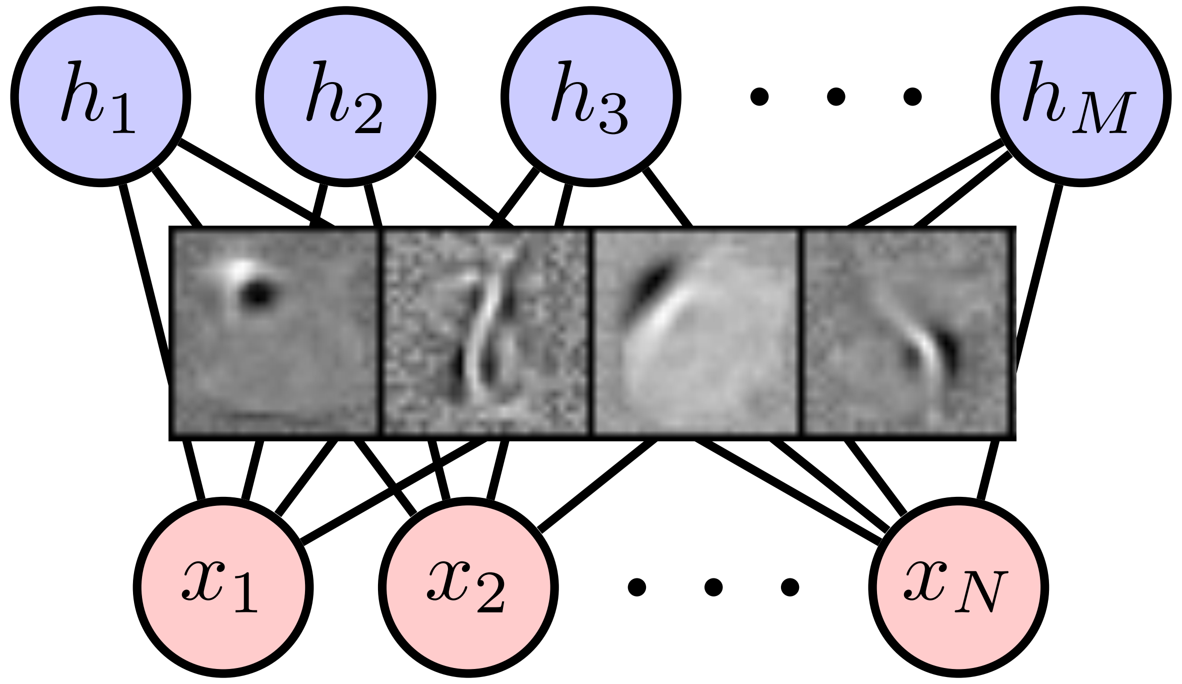
Fisher divergence, defined as

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

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

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

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]

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

Speech: WaveNet 1609.03499

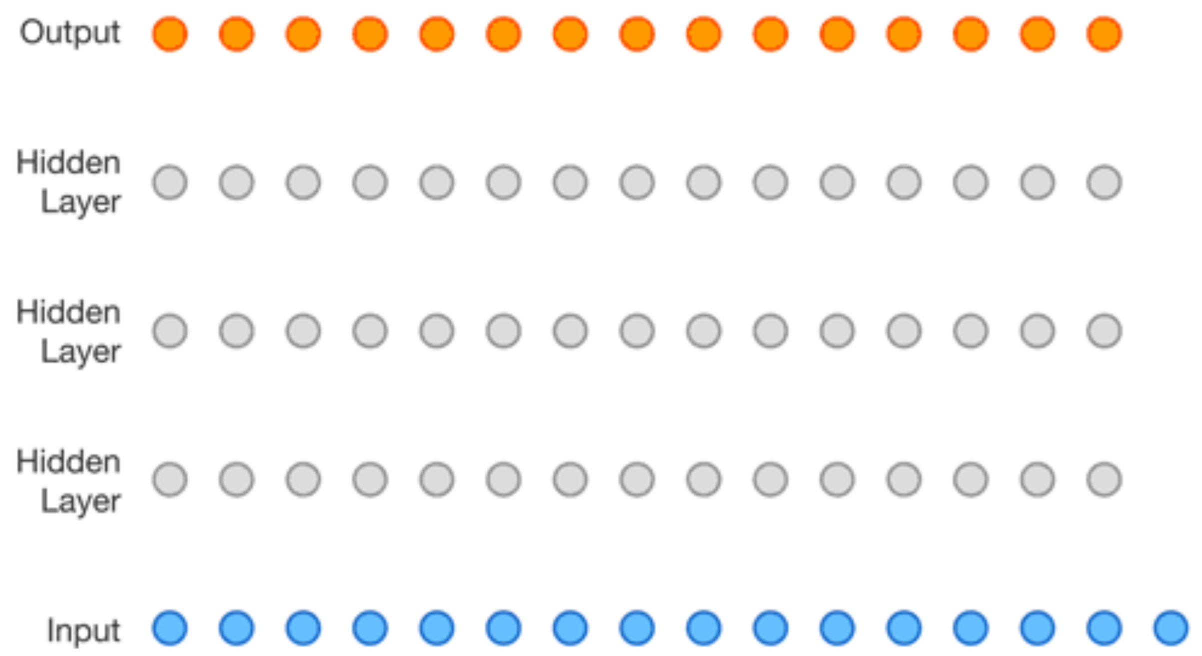
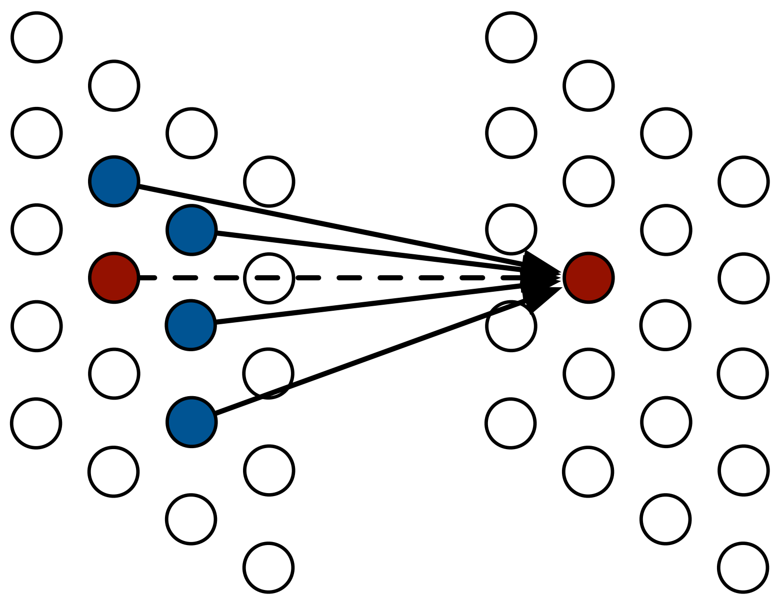
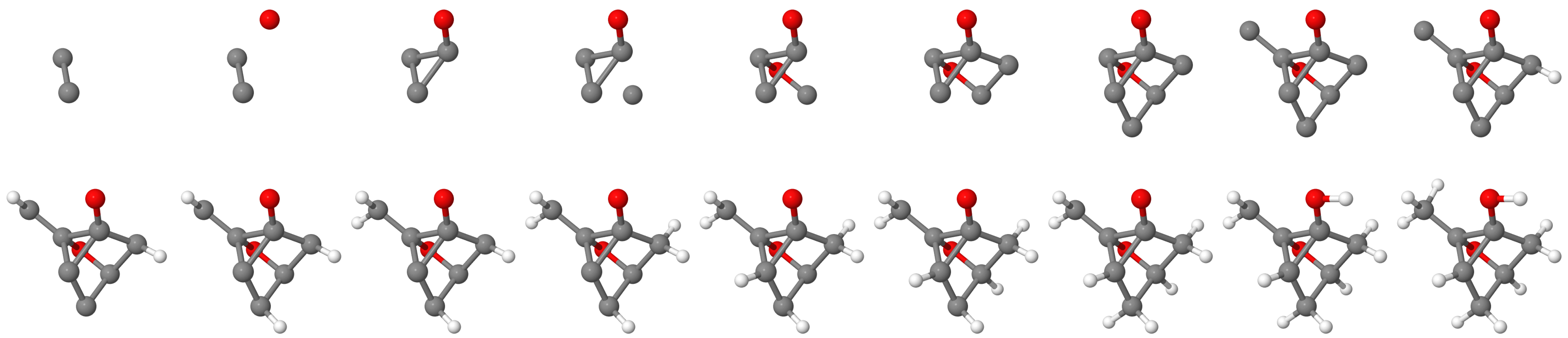


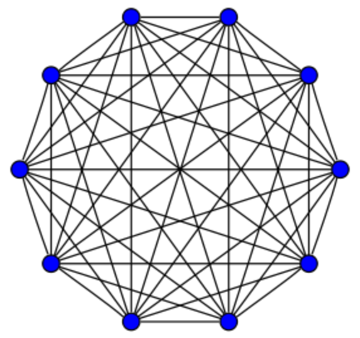
Image: PixelCNN 1601.06759



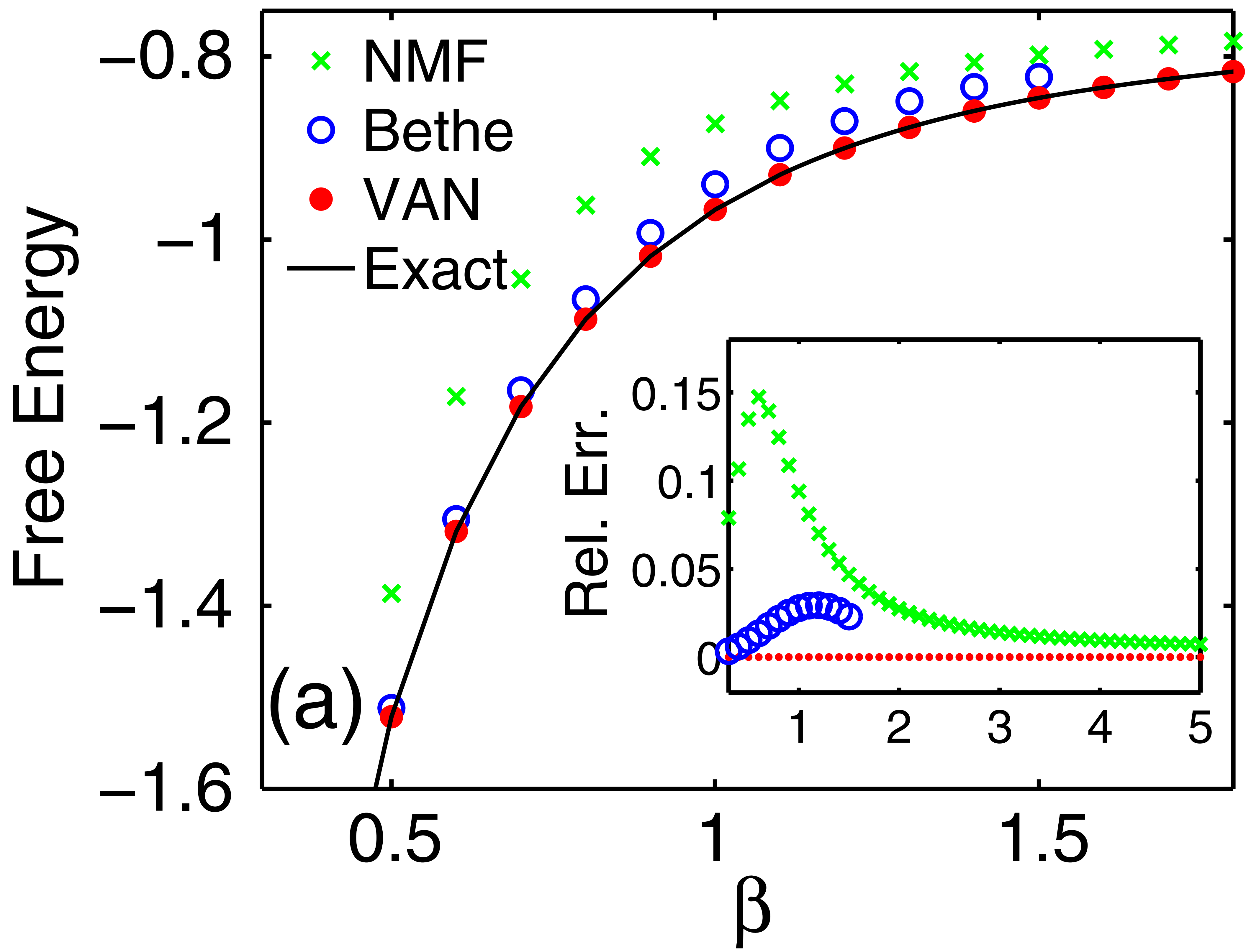
Molecular graph: 1810.11347



Variational autoregressive networks



Sherrington-Kirkpatrick spin glass



Variational autoregressive network

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

github.com/wdphy16/stat-mech-van

Wu, LW, Zhang, PRL '19

Conventional approaches

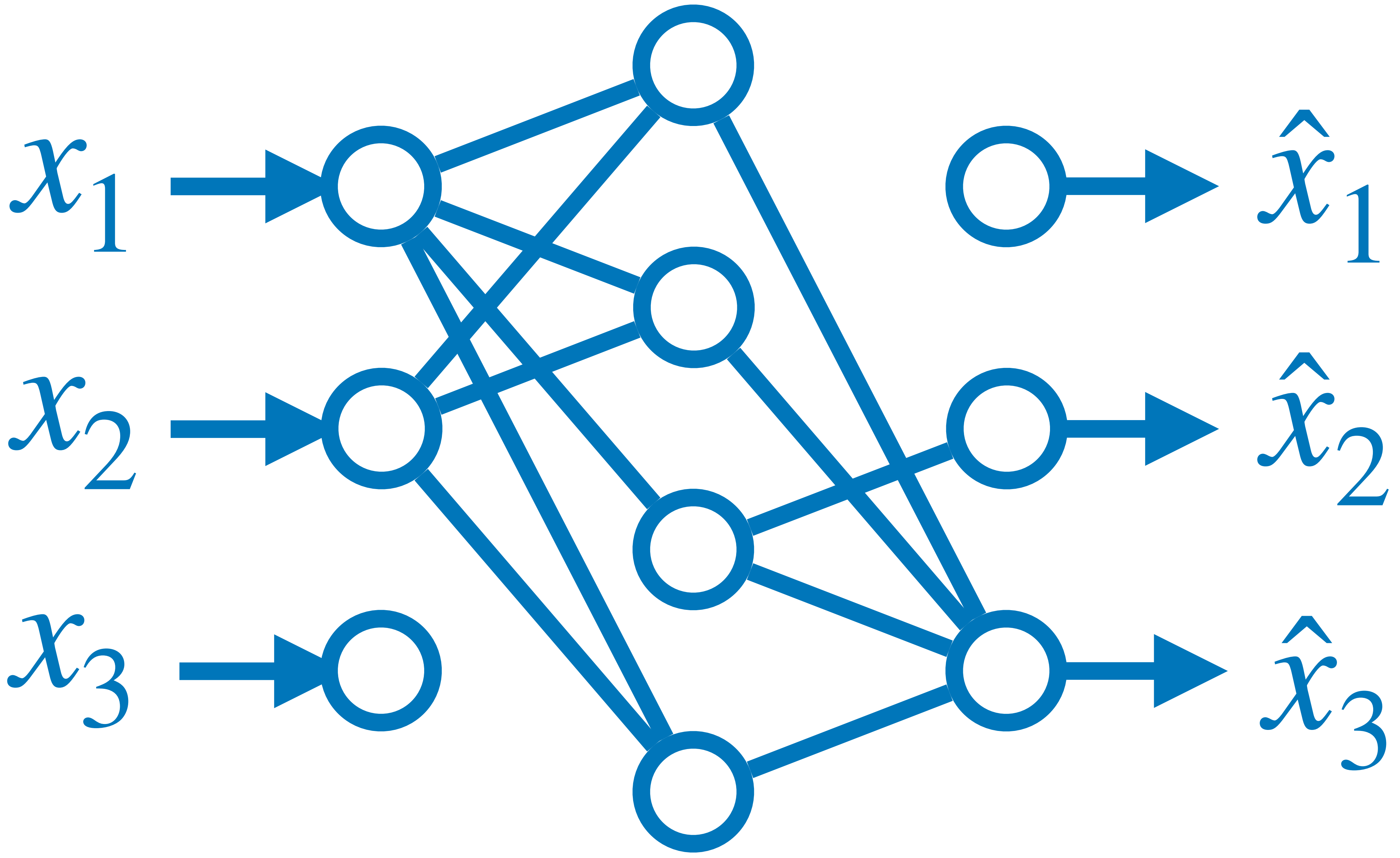
Naive mean-field
factorized probability

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

Bethe approximation
pairwise interaction

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

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

Normalizing flows

Parallel WaveNet 1711.10433

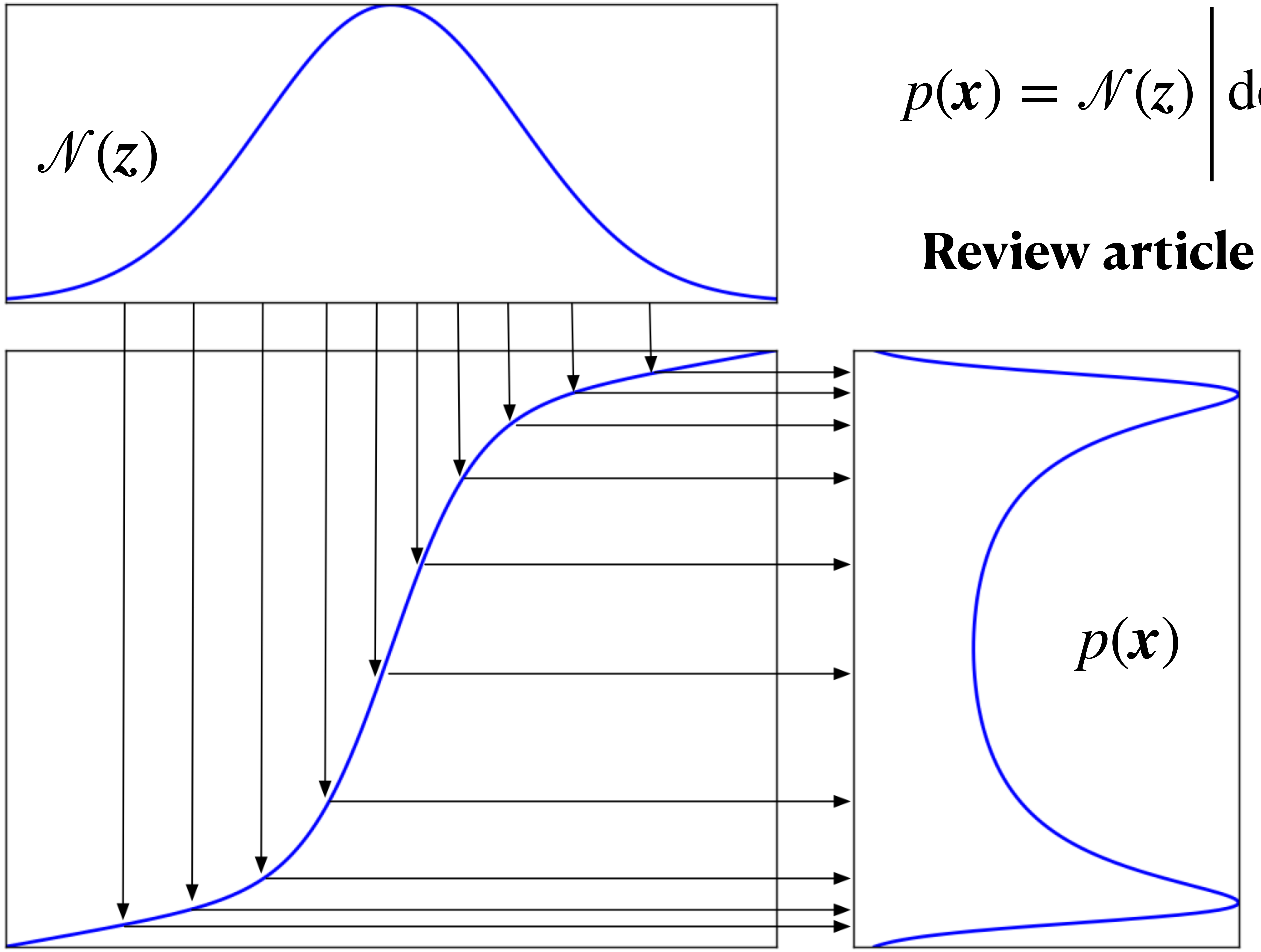
https://deepmind.com/blog/high-fidelity-speech-synthesis-wavenet/

Glow 1807.03039

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

Normalizing flow in a nutshell

Base density



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

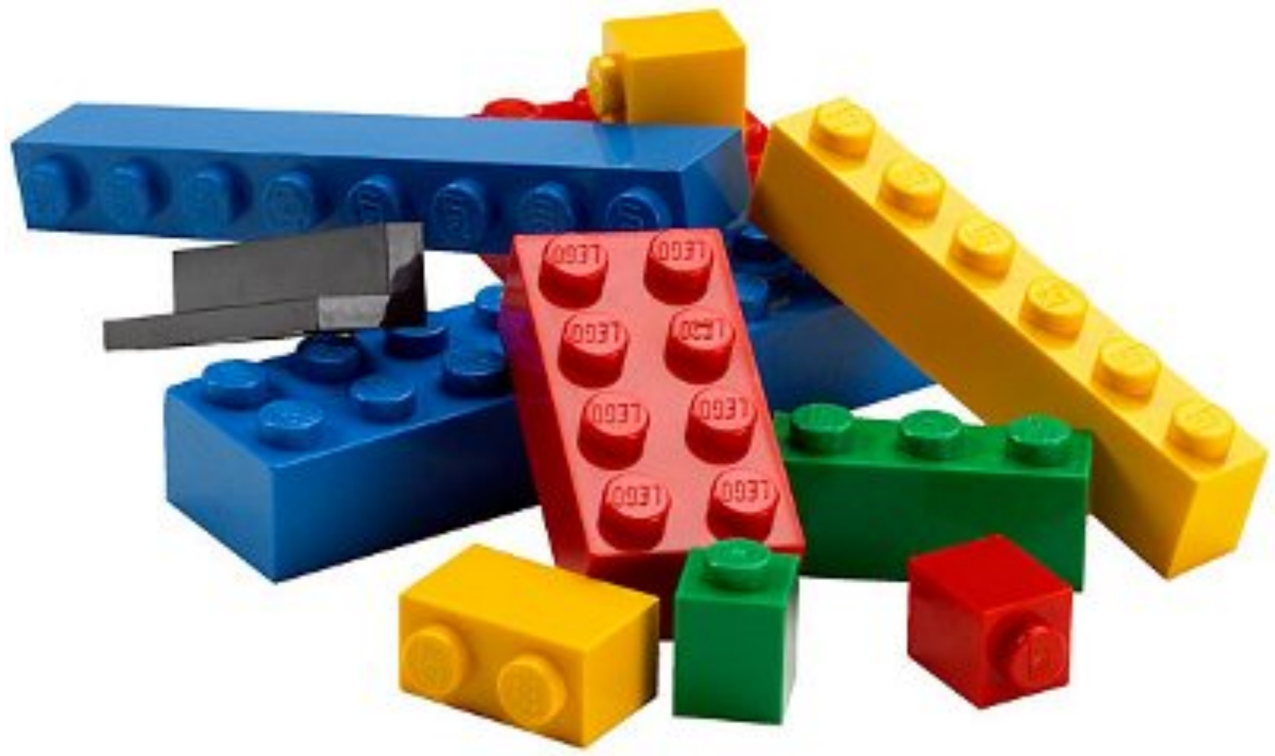
Review article 1912.02762

"neural net" with 1 neuron

Target density

Flow architecture design

Composability

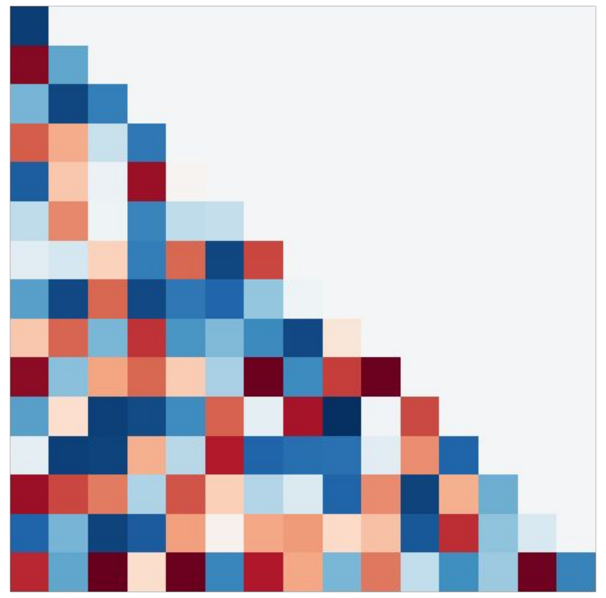


$$z = \mathcal{T}(\boldsymbol{x})$$

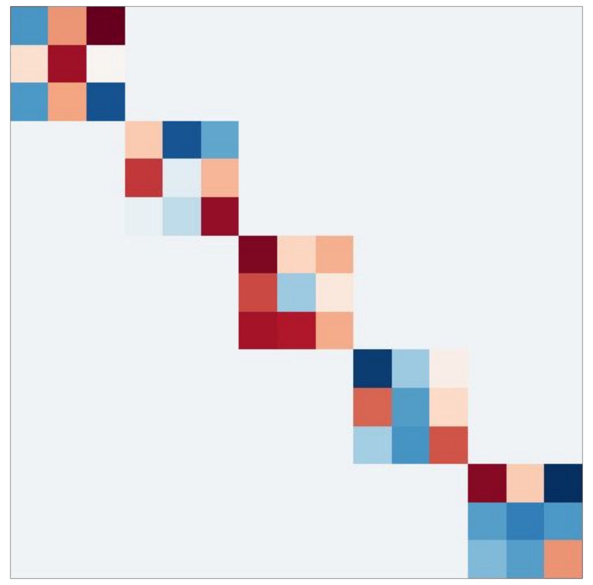
$$\mathcal{T} = \mathcal{T}_1 \circ \mathcal{T}_2 \circ \mathcal{T}_3 \circ \cdots$$

Balanced efficiency & inductive bias

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



Autoregressive



Neural RG

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

Continuous flow

Example of a building block

Forward

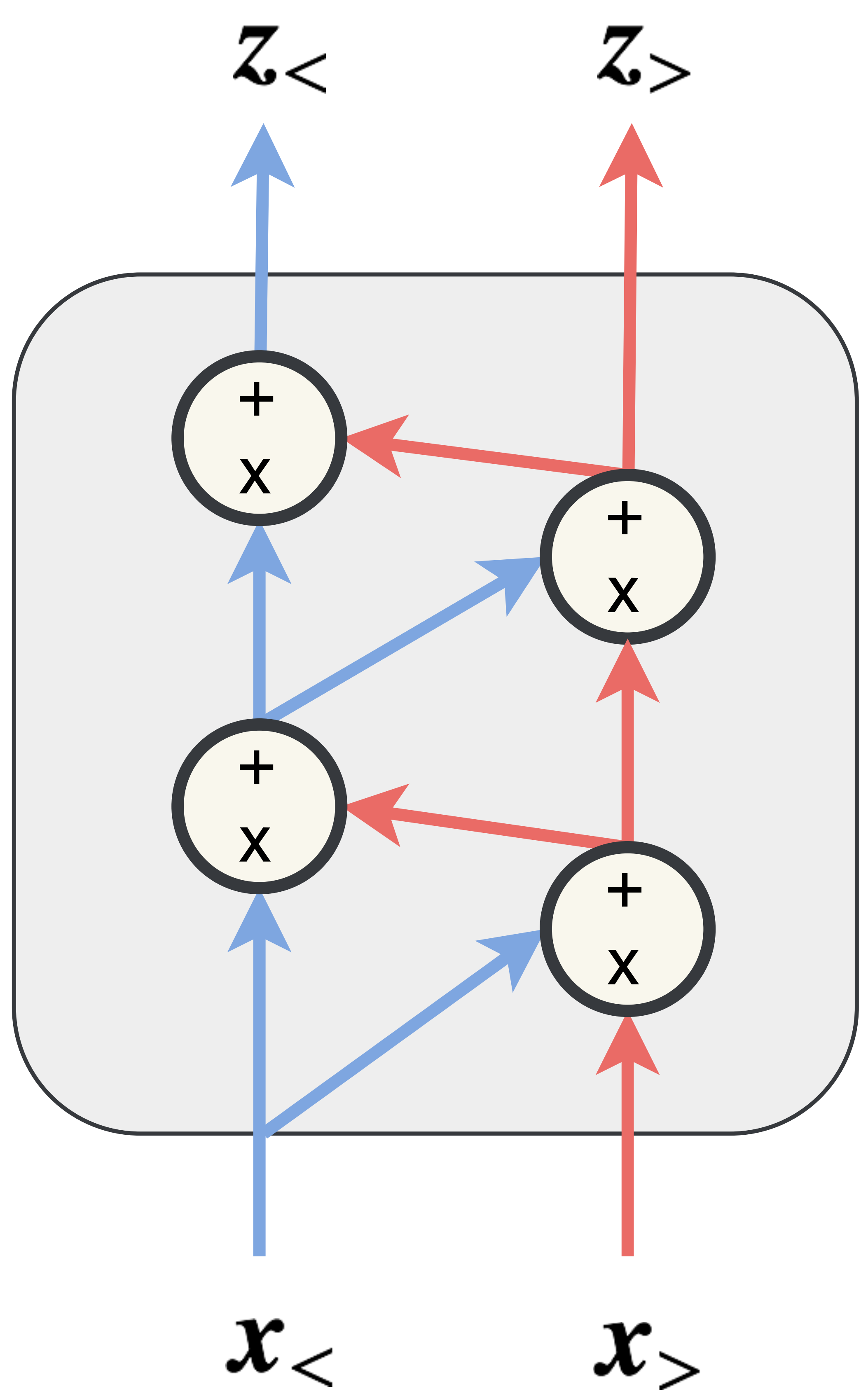
$$\begin{cases} x_< = z_< \\ x_> = z_> \odot e^{s(z_<)} + t(z_<) \end{cases}$$

Inverse

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

Log-Abs-Jacobian-Det

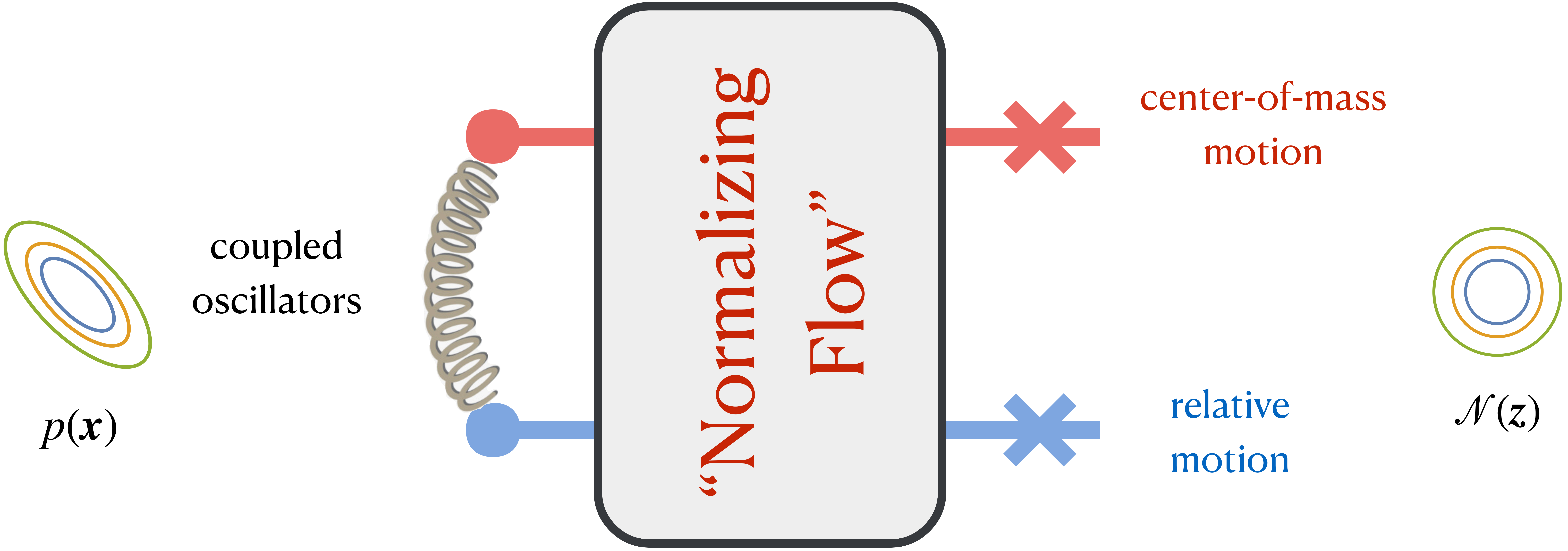
$$\ln \left| \det \left(\frac{\partial x}{\partial z} \right) \right| = \sum_i [s(z_<)]_i$$



Real NVP, Dinh et al,1605.08803

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

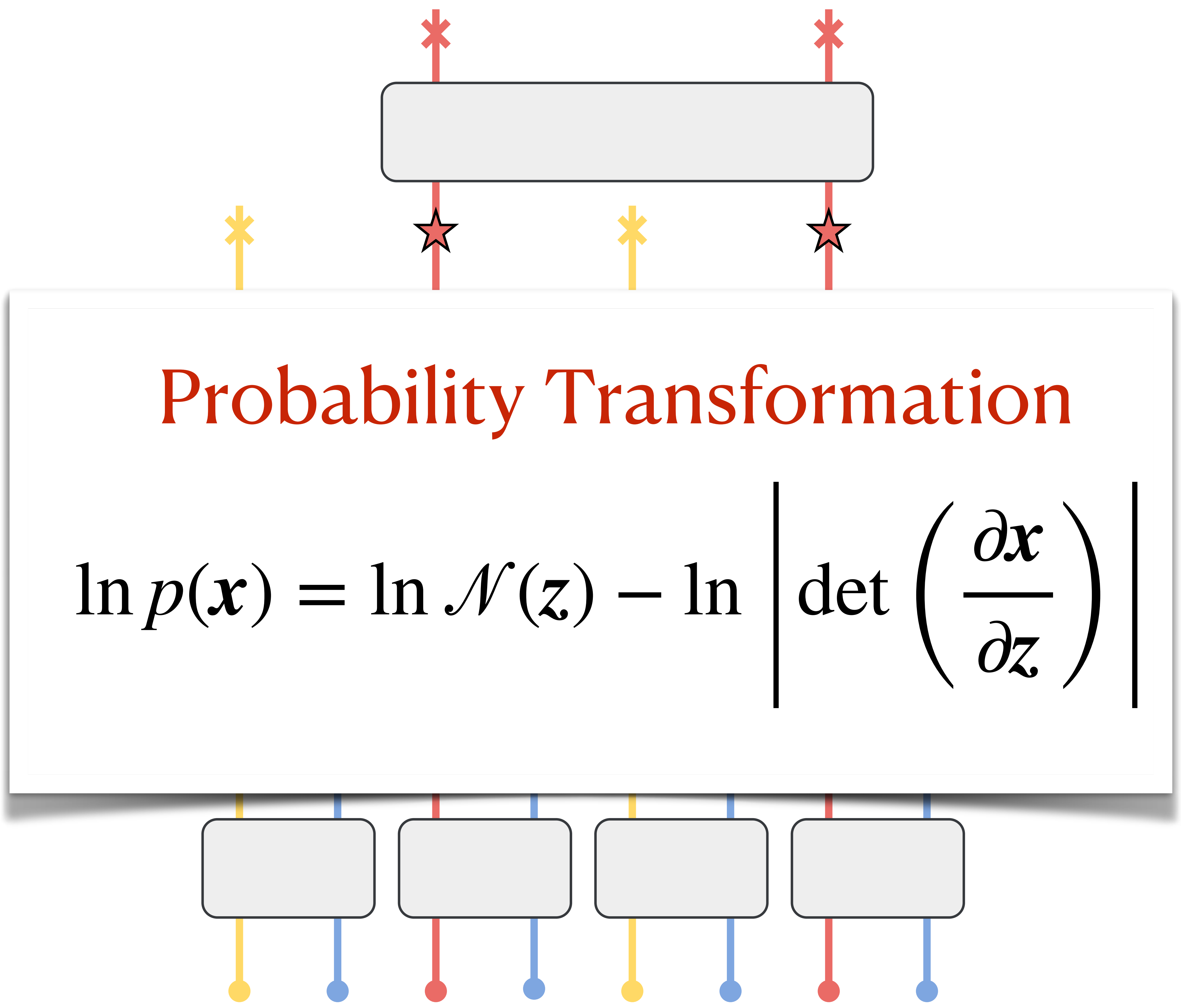
Normalizing flow for physics: an intuition



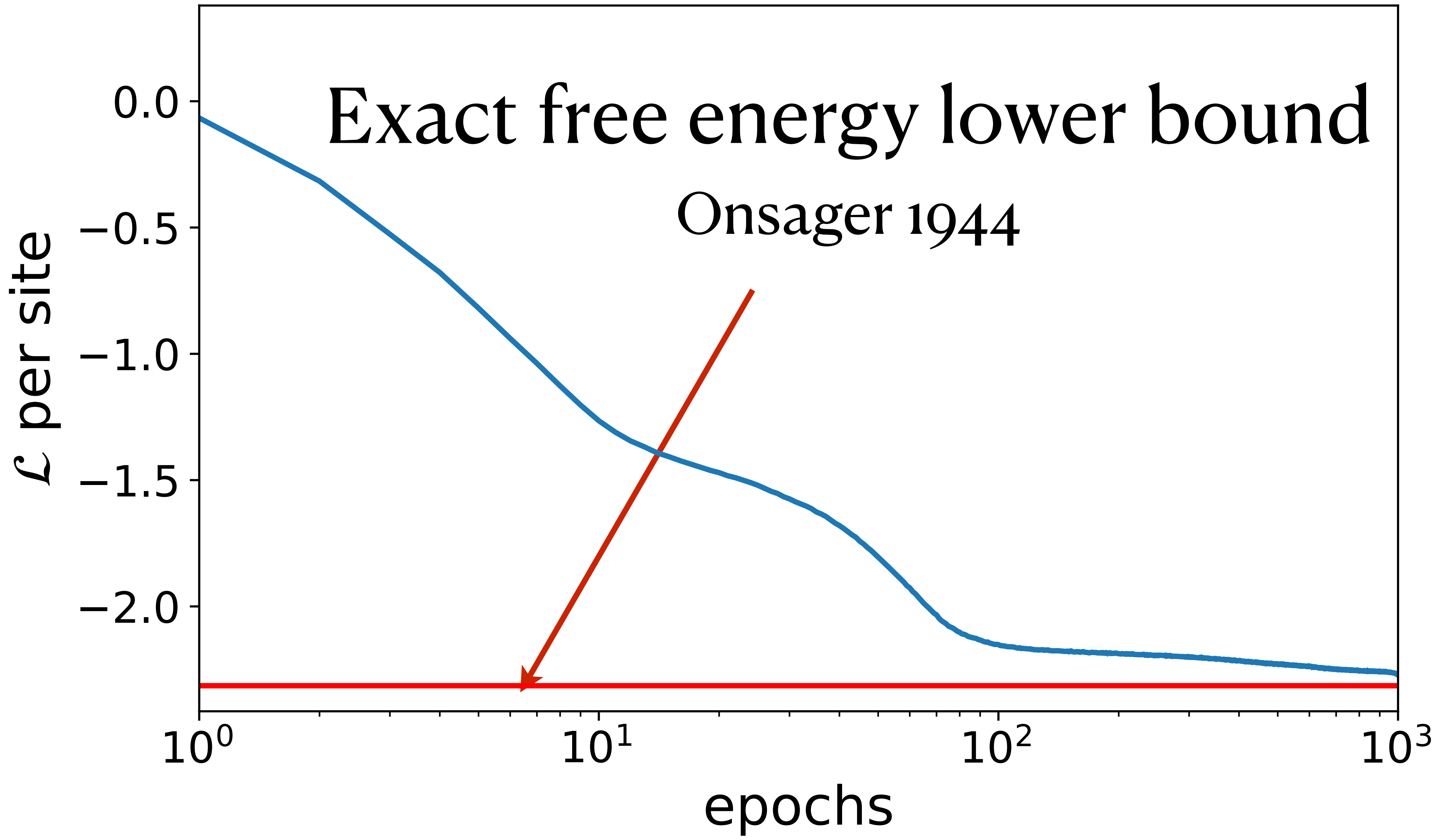
High-dimensional, composable, learnable, nonlinear transformations

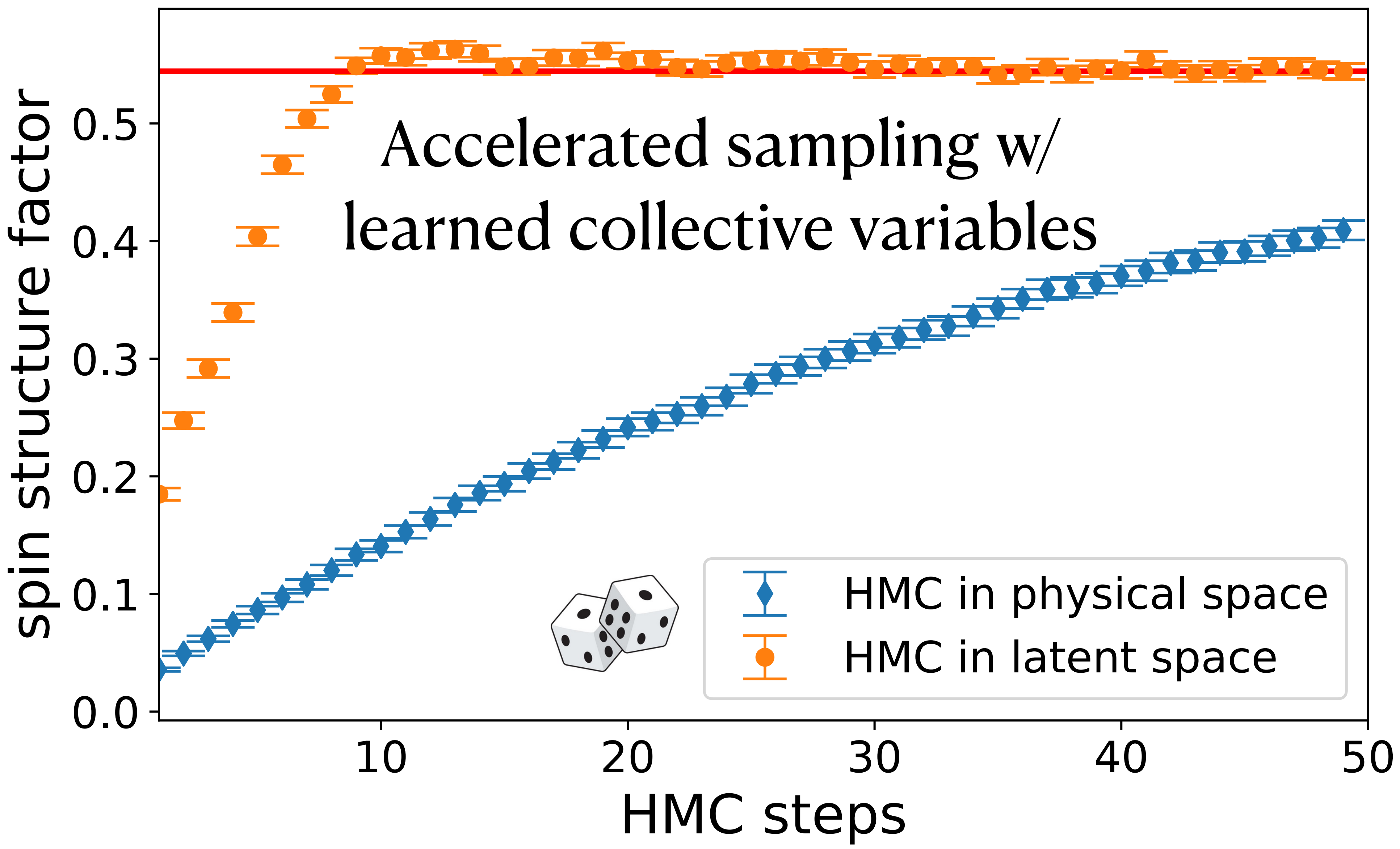
Neural network renormalization group

Li, LW, PRL '18 [li012589/NeuralRG]

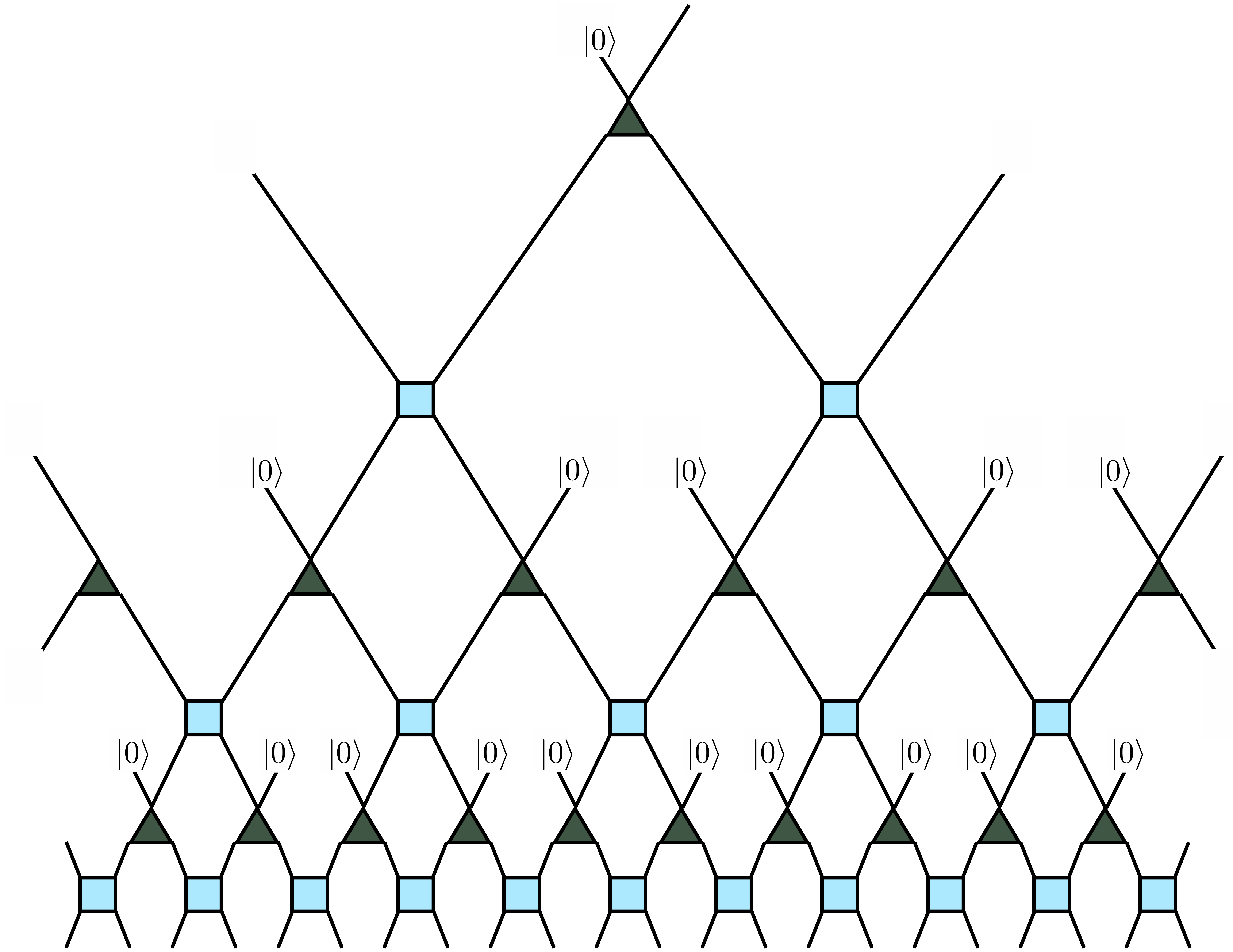


$$\ln p(\boldsymbol{x}) = \ln \mathcal{N}(\boldsymbol{z}) - \ln \left| \det \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} \right) \right|$$

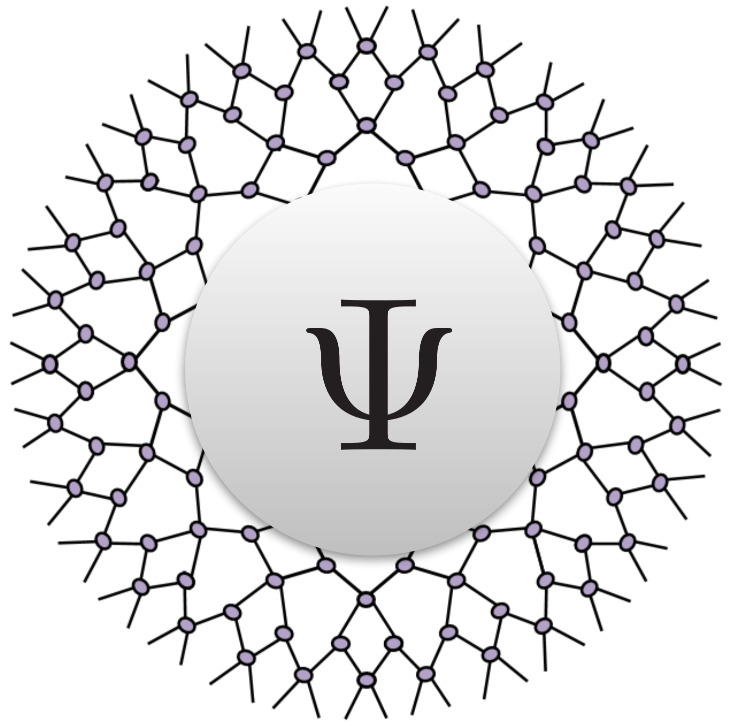




Quantum version of the architecture

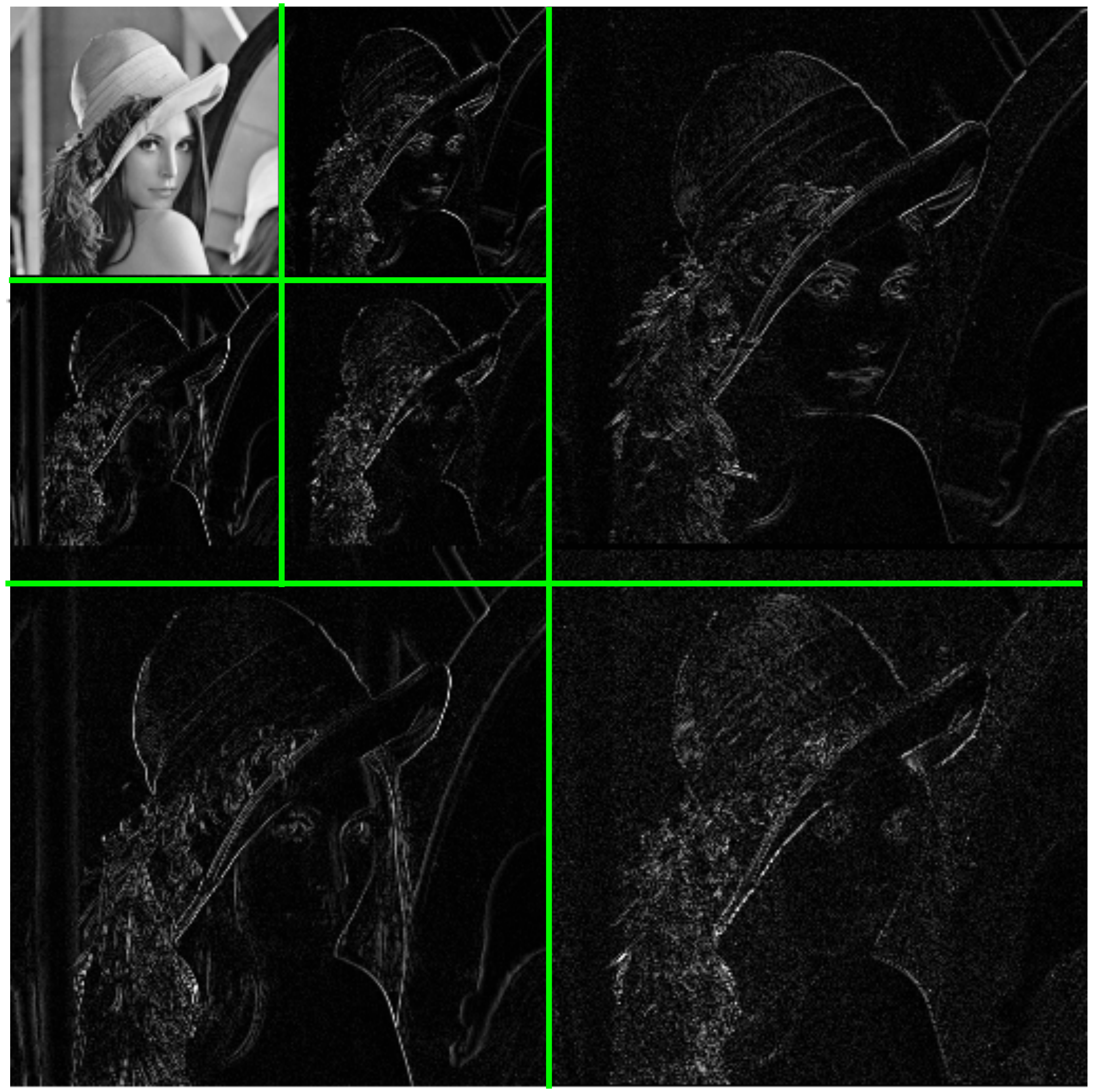


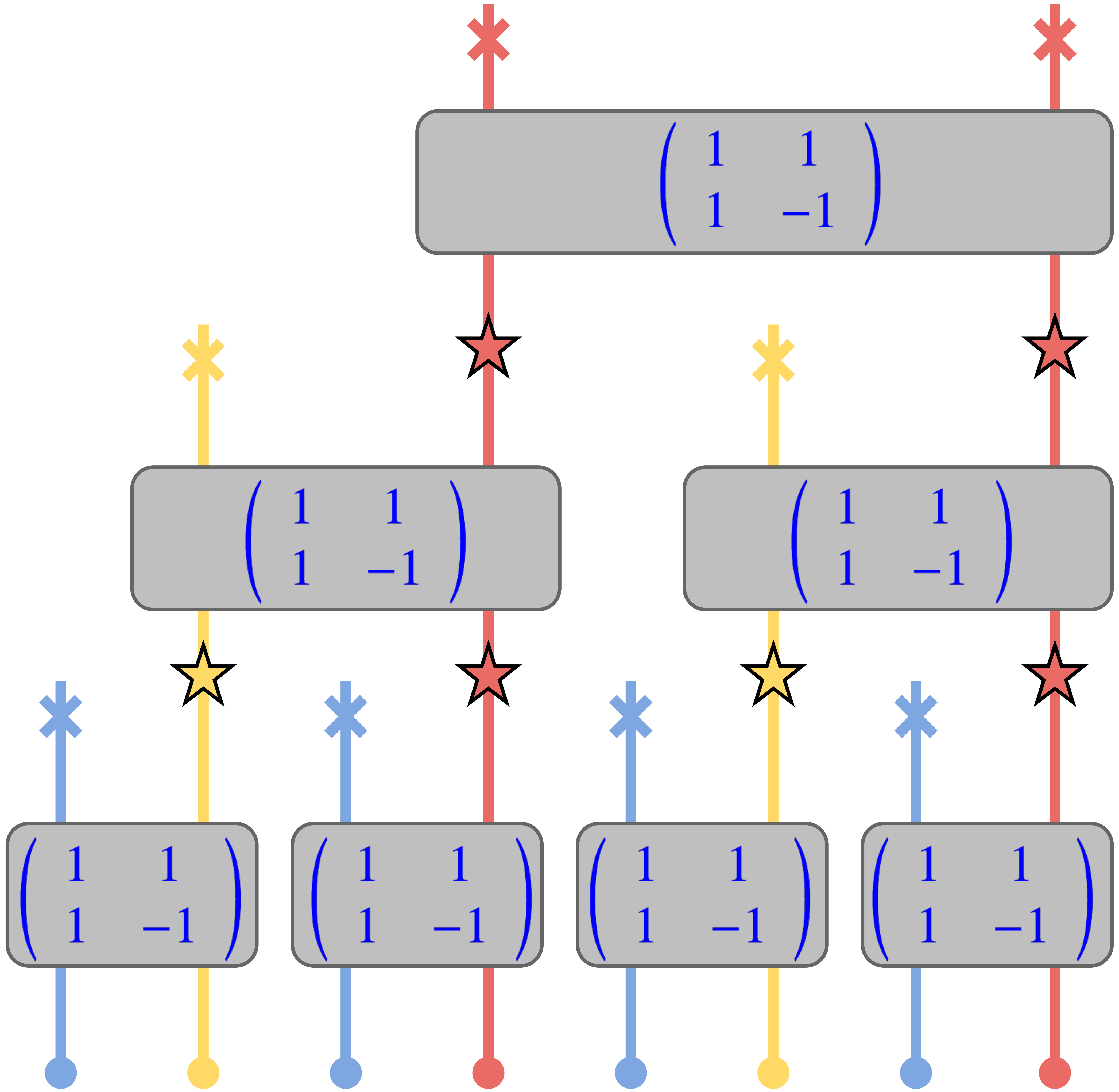
Entangled qubits



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+

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

$$\boldsymbol{x} = \boldsymbol{z} + \varepsilon \boldsymbol{v}$$

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

$\varepsilon \to 0$

$t = 1$ $t = 0$

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

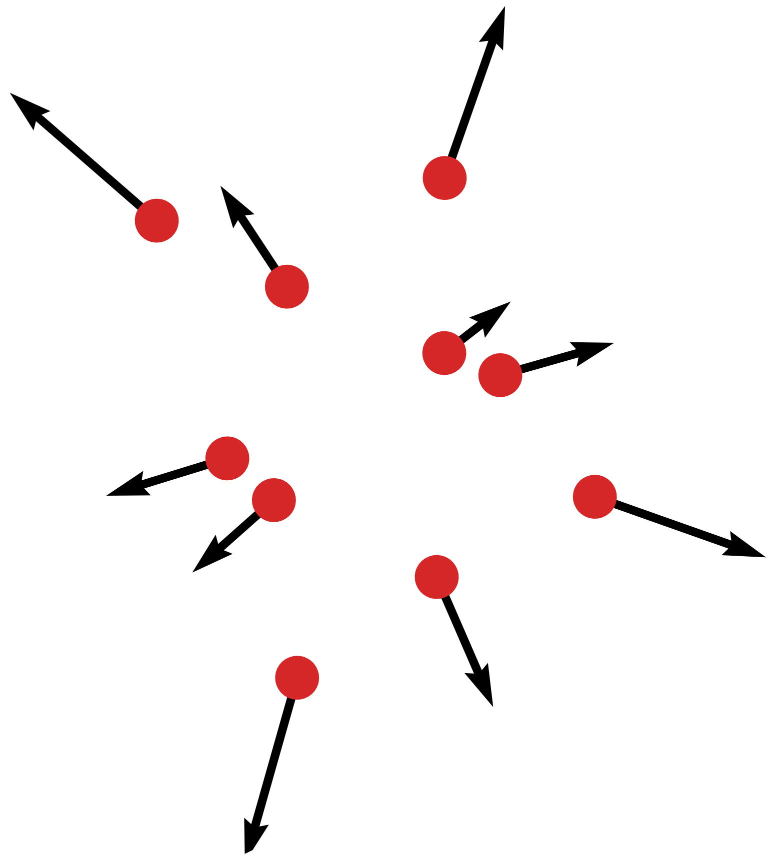
$$\frac{d \ln p(\boldsymbol{x}, t)}{dt} = -\nabla \cdot \boldsymbol{v}$$

Fluid physics behind flows

Zhang, E, LW 1809.10188
[wangleiphy/MongeAmpereFlow](https://github.com/wangleiphy/MongeAmpereFlow)

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

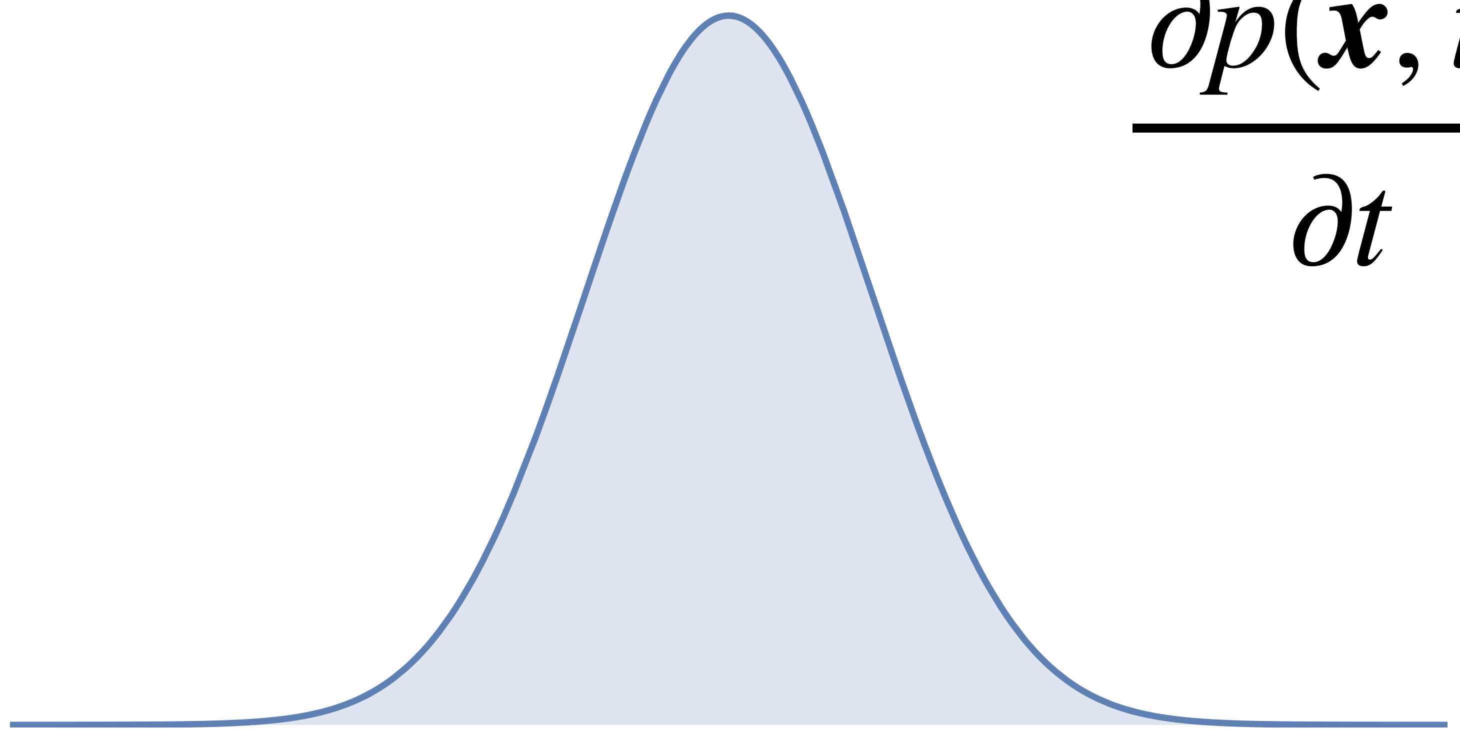
$$\frac{d \ln p(\boldsymbol{x}, t)}{dt} = -\nabla \cdot \boldsymbol{v}$$



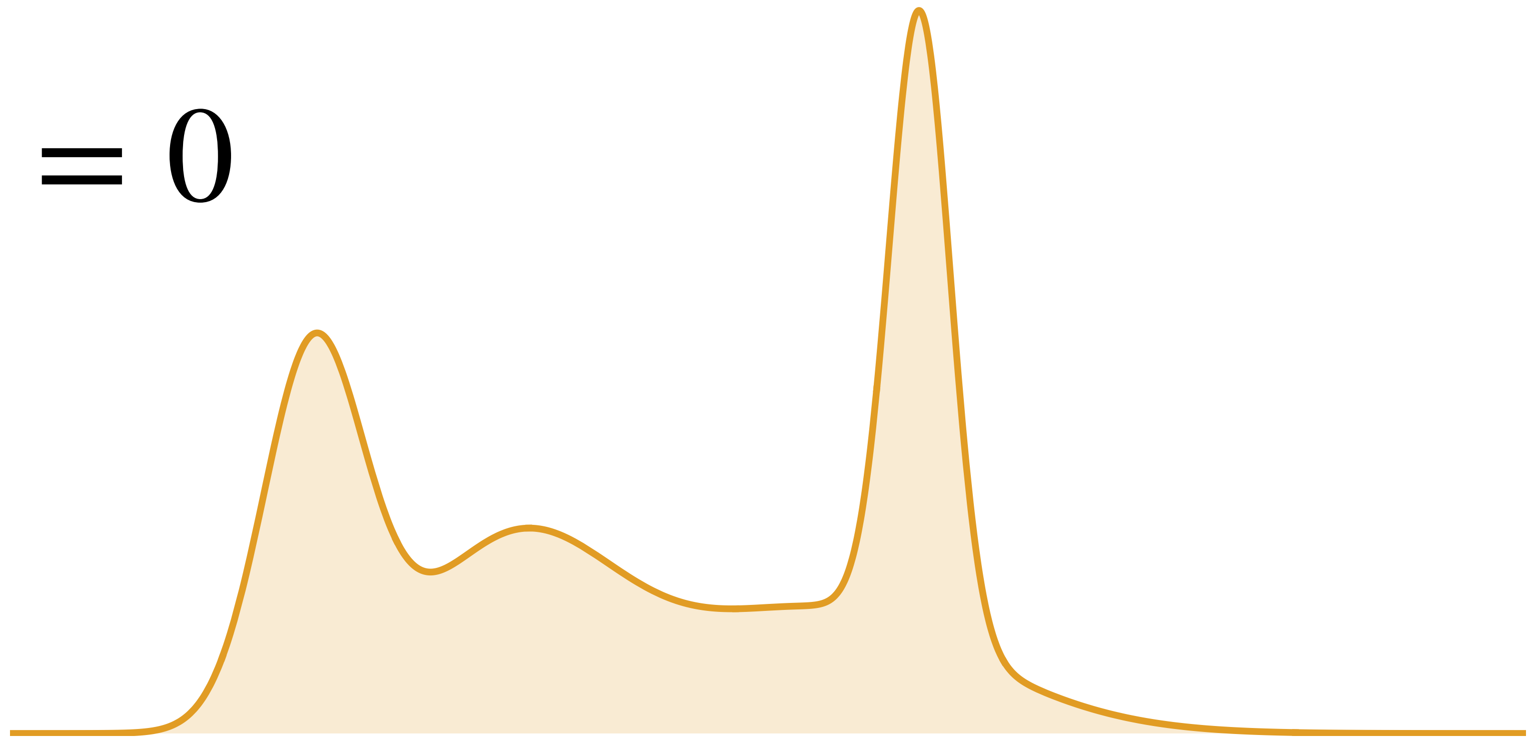
$$\frac{d}{dt} = \frac{\partial}{\partial t} + \boldsymbol{v} \cdot \nabla$$

“material derivative”

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



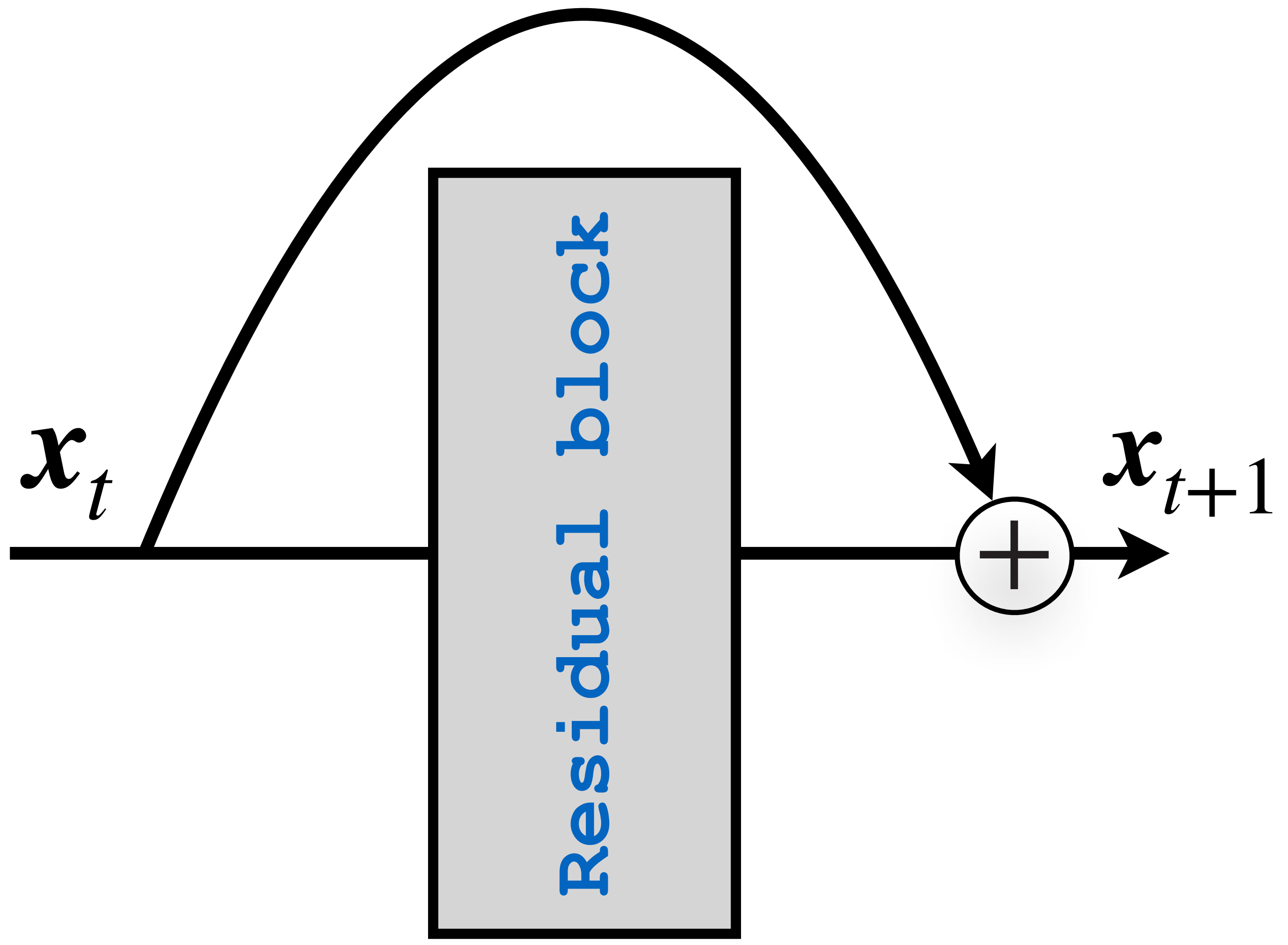
Simple density



Complex density

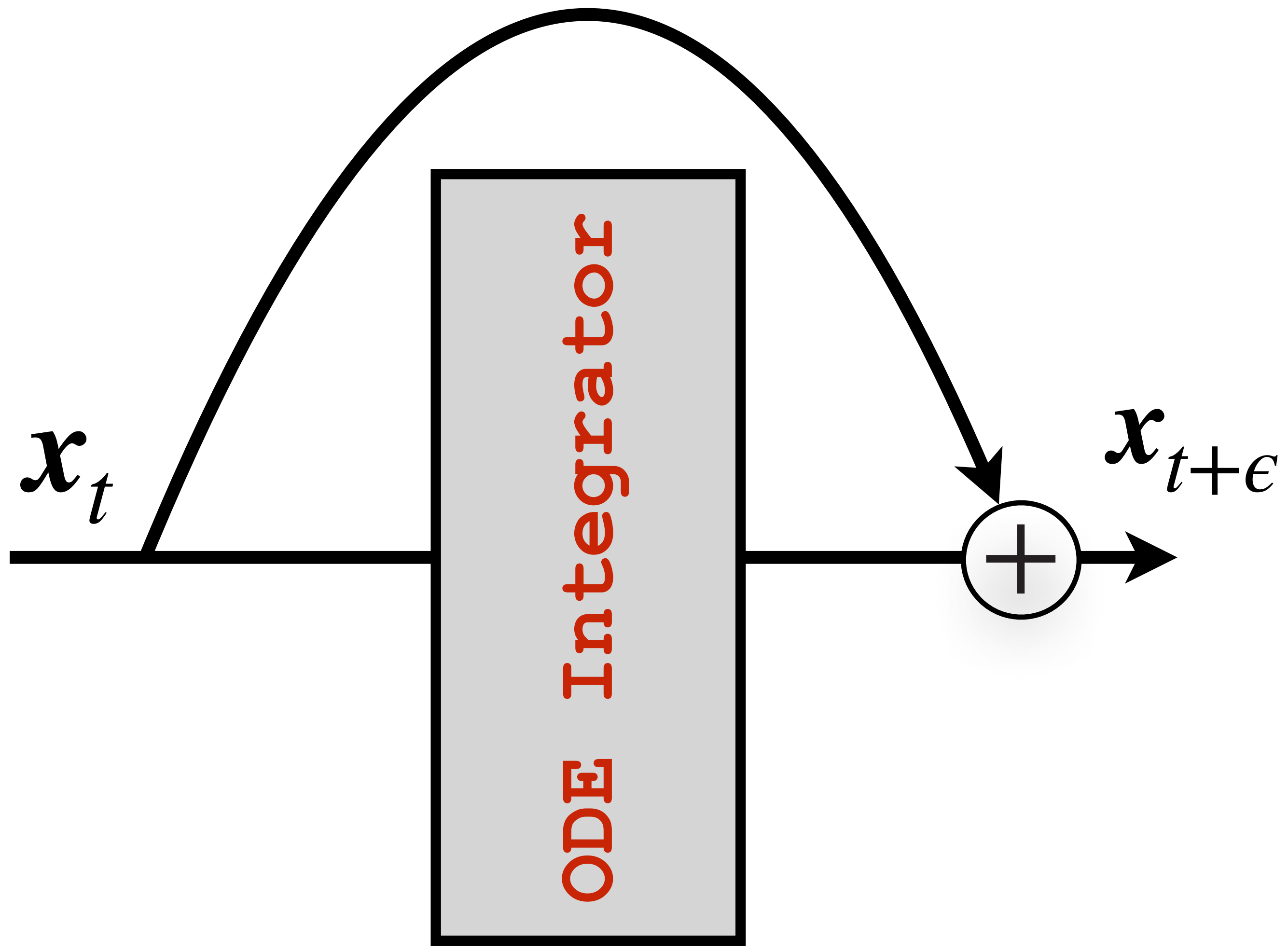
Neural Ordinary Differential Equations

Residual network



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

ODE integration



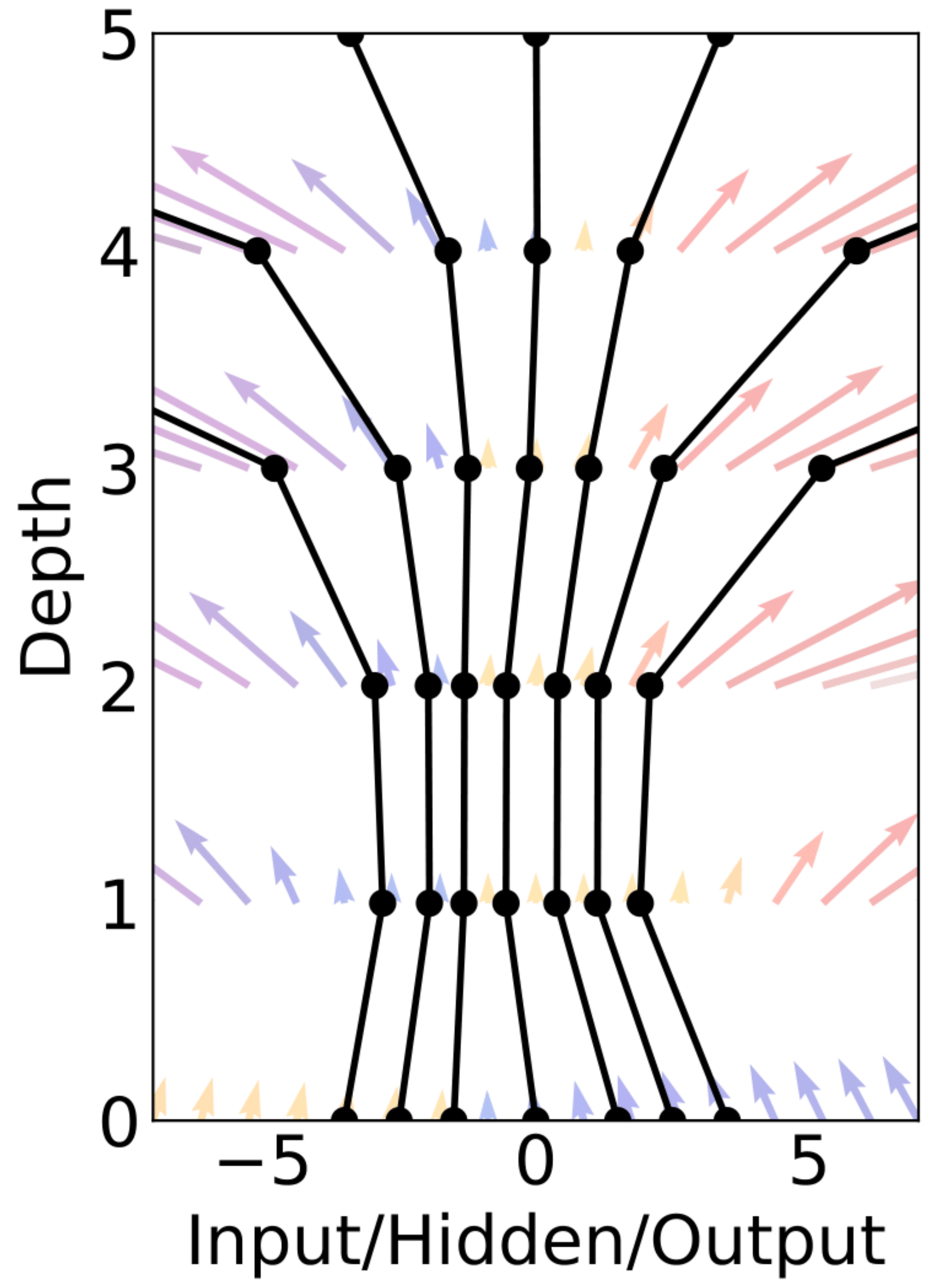
$$dx/dt = v(x)$$

Chen et al, 1806.07366

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

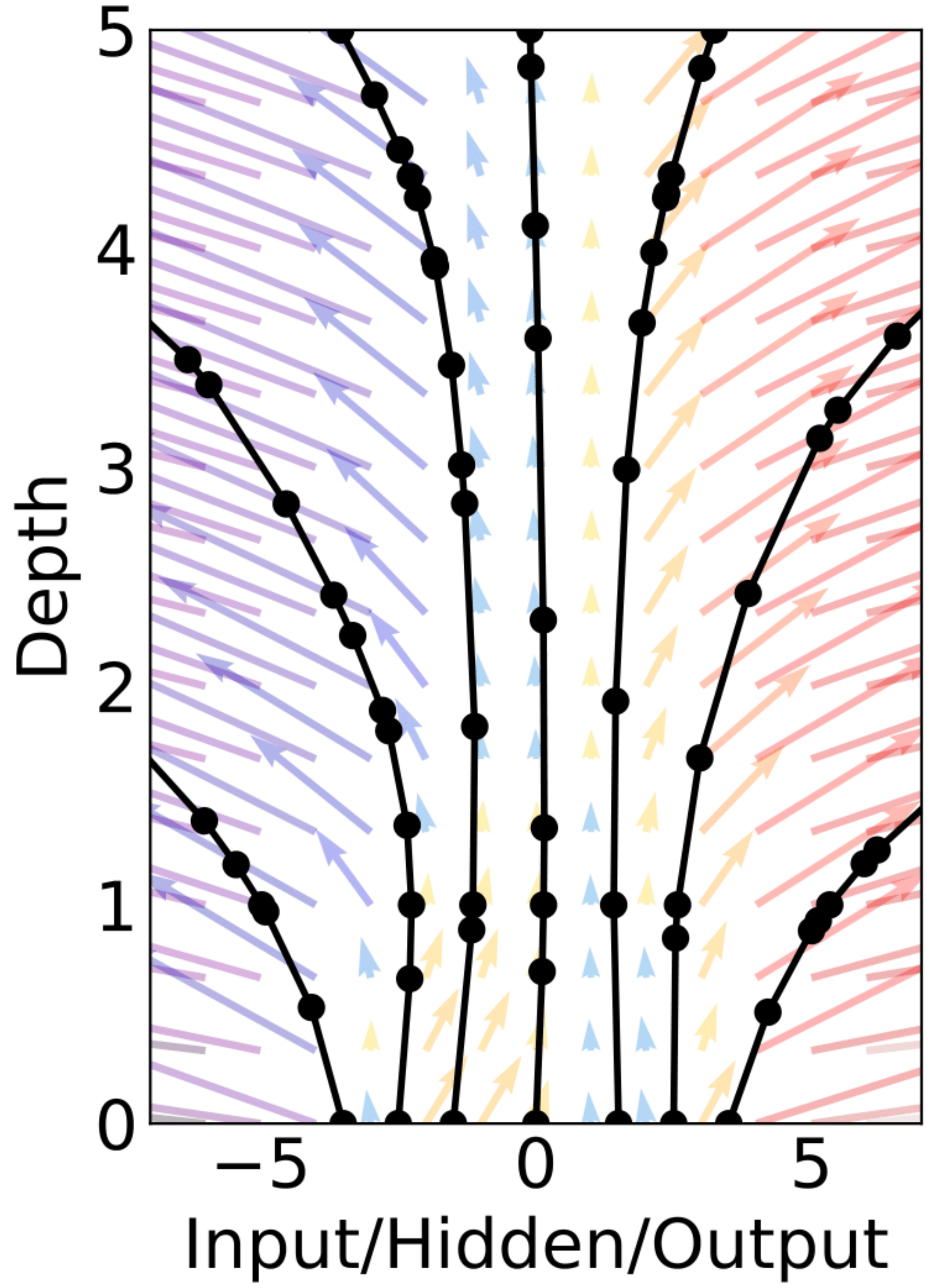
Neural Ordinary Differential Equations

Residual network



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

ODE integration



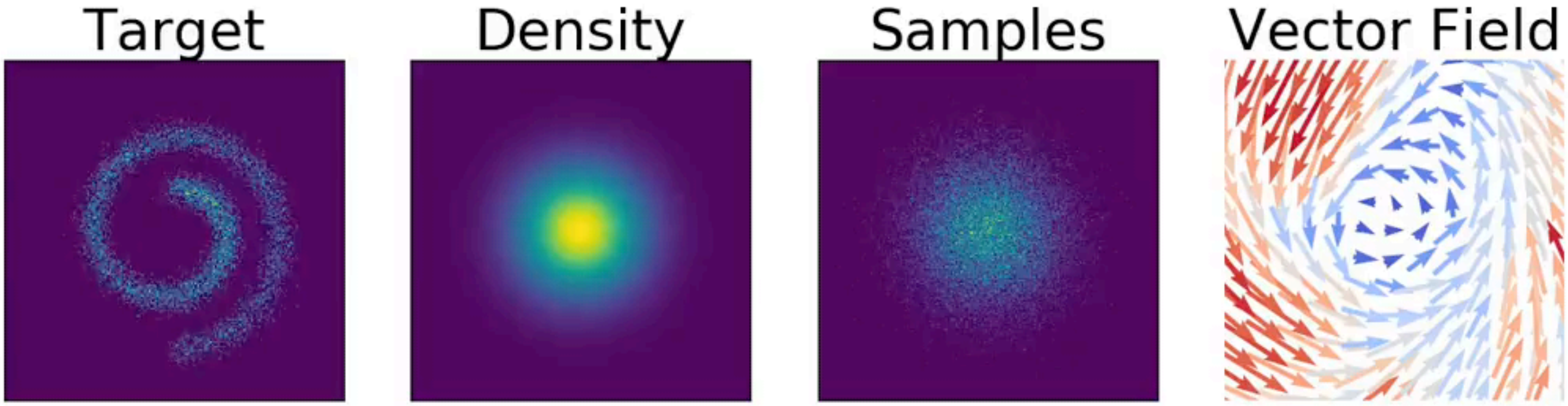
$$dx/dt = v(x)$$

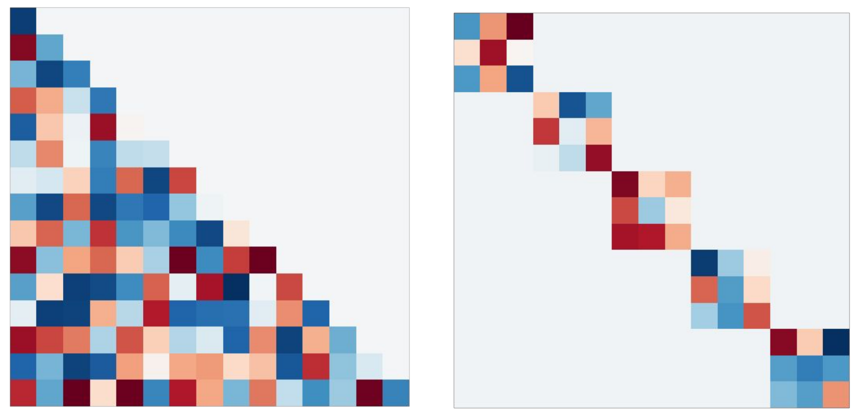
Chen et al, 1806.07366

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

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

(a) Density estimation

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

“learn from data”

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

(b) Variational free energy

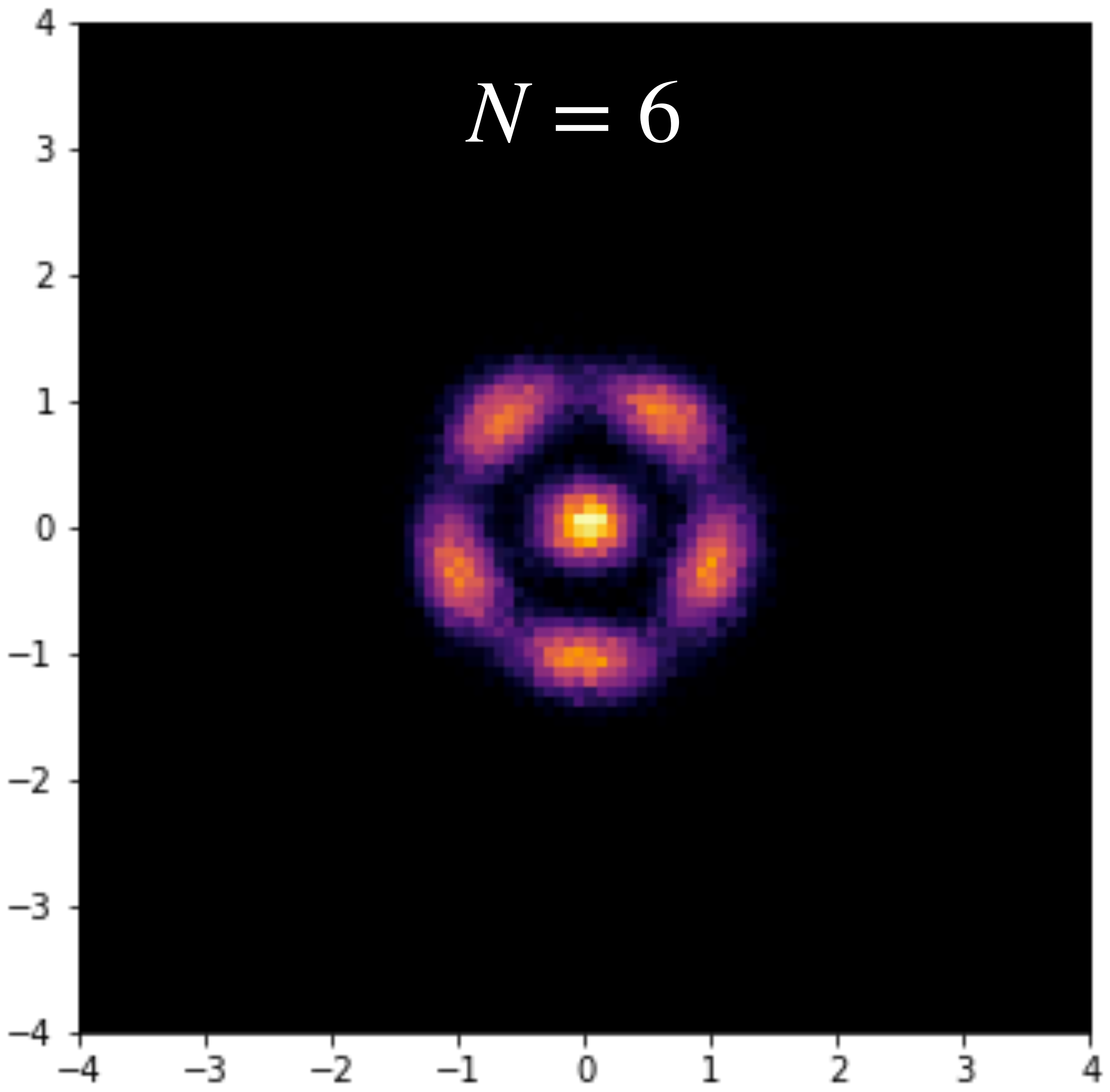
$$\mathcal{N}(\boldsymbol{x}) \xrightarrow{\int_0^T \dot{\boldsymbol{x}}\, dt} \frac{e^{-E(\boldsymbol{x})}}{Z}$$

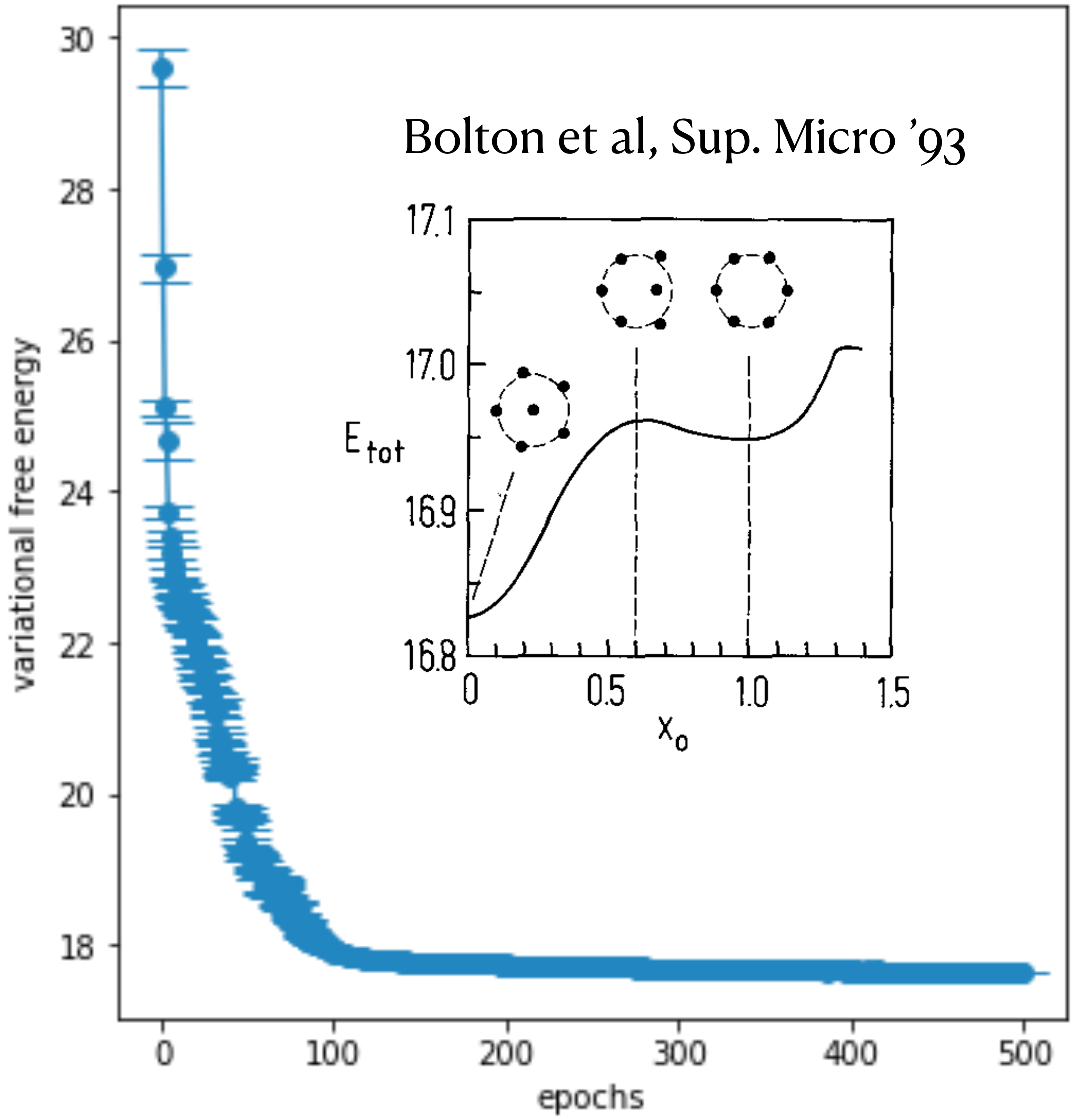
“learn from Hamiltonian”

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

Demo: 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

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

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

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

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

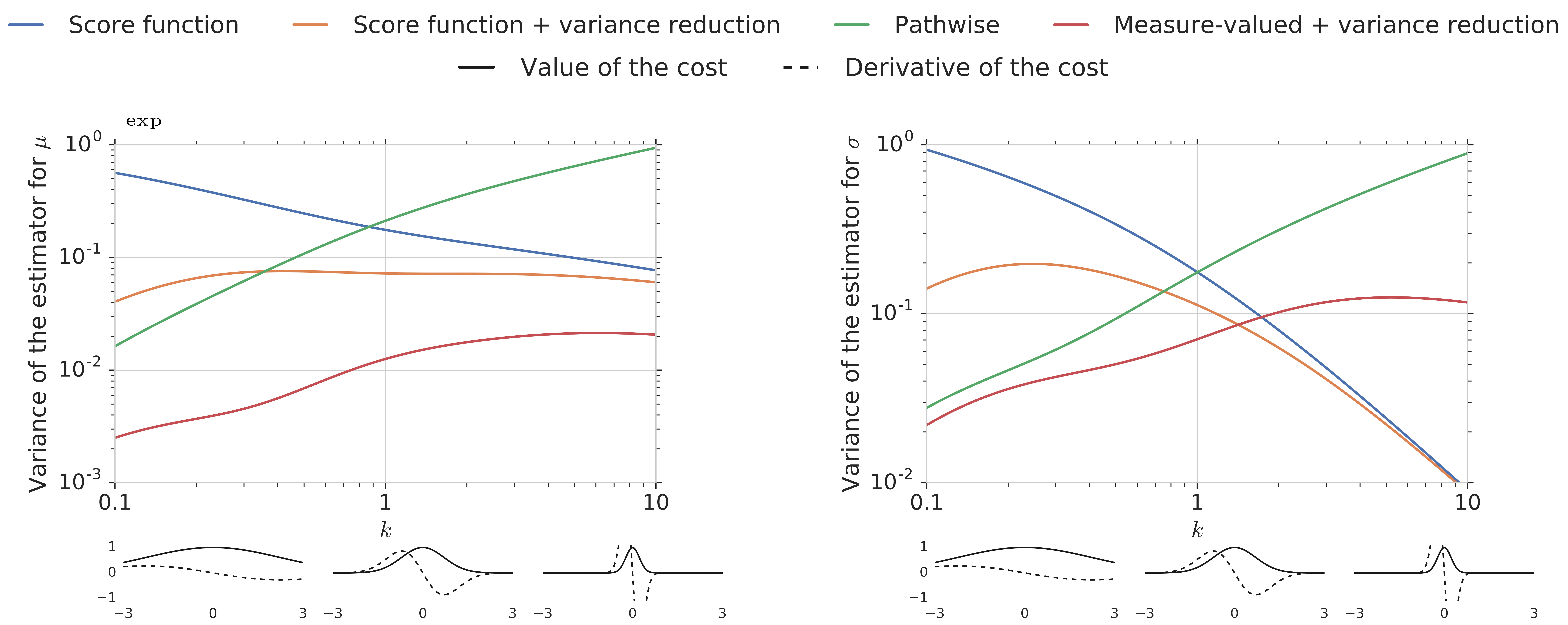
When to use which ?

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

A few words about tooling

HIPS/autograd

theano

Zygote

MindSpore

PyTorch

TensorFlow

SciML

Keras

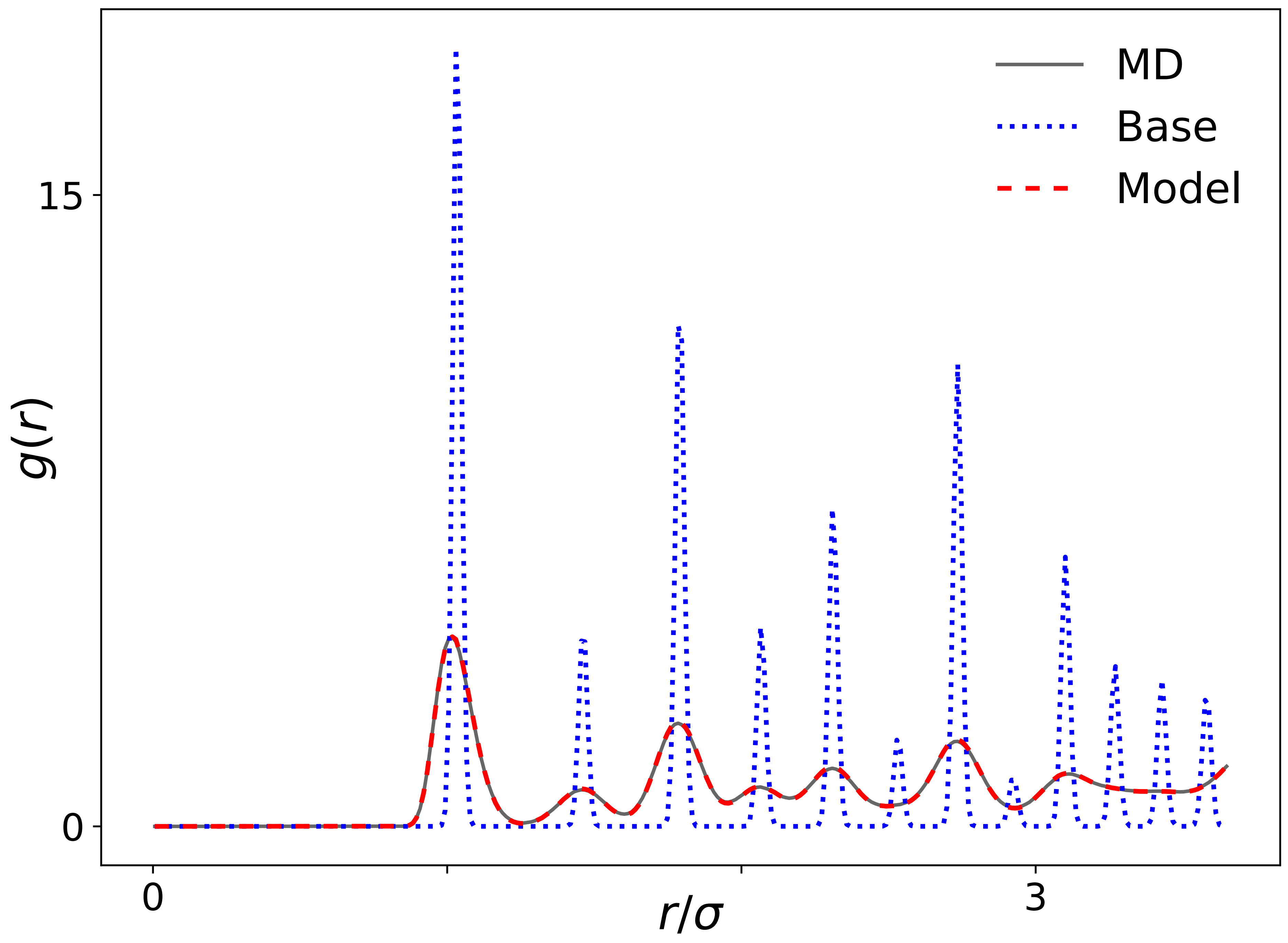
NiLang

Taichi

Differentiable programming frameworks

Case study: Normalizing flow for atomic solids

Variational free energy with a really deep (and a bit awkward) 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 preserving permutation. But, does it really matter?

Diffusion models

Denoising score matching Vincent 2011

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}[\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\|_2^2].$$

Diffusion generative model

Sohl-Dickstein et al, 1503.03585

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$$

Song et al, 1907.05600, Ho et al, 2006.11239

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

A tale of three equations

Langevin equation (SDE)

$$q(\boldsymbol{x}_{t+dt}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_t + \boldsymbol{f}dt, 2Tdt\boldsymbol{I}) \quad \text{or} \quad \boldsymbol{x}_{t+dt} - \boldsymbol{x}_t = \boldsymbol{f}dt + \sqrt{2Tdt}\,\mathcal{N}(0,1)$$

Fokker-Planck equation (PDE)

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

"Particle method" (ODE)

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

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

Maoutsa et al, 2006.00702
Song et al, 2011.13456

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

and simplified by the change of variables,

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

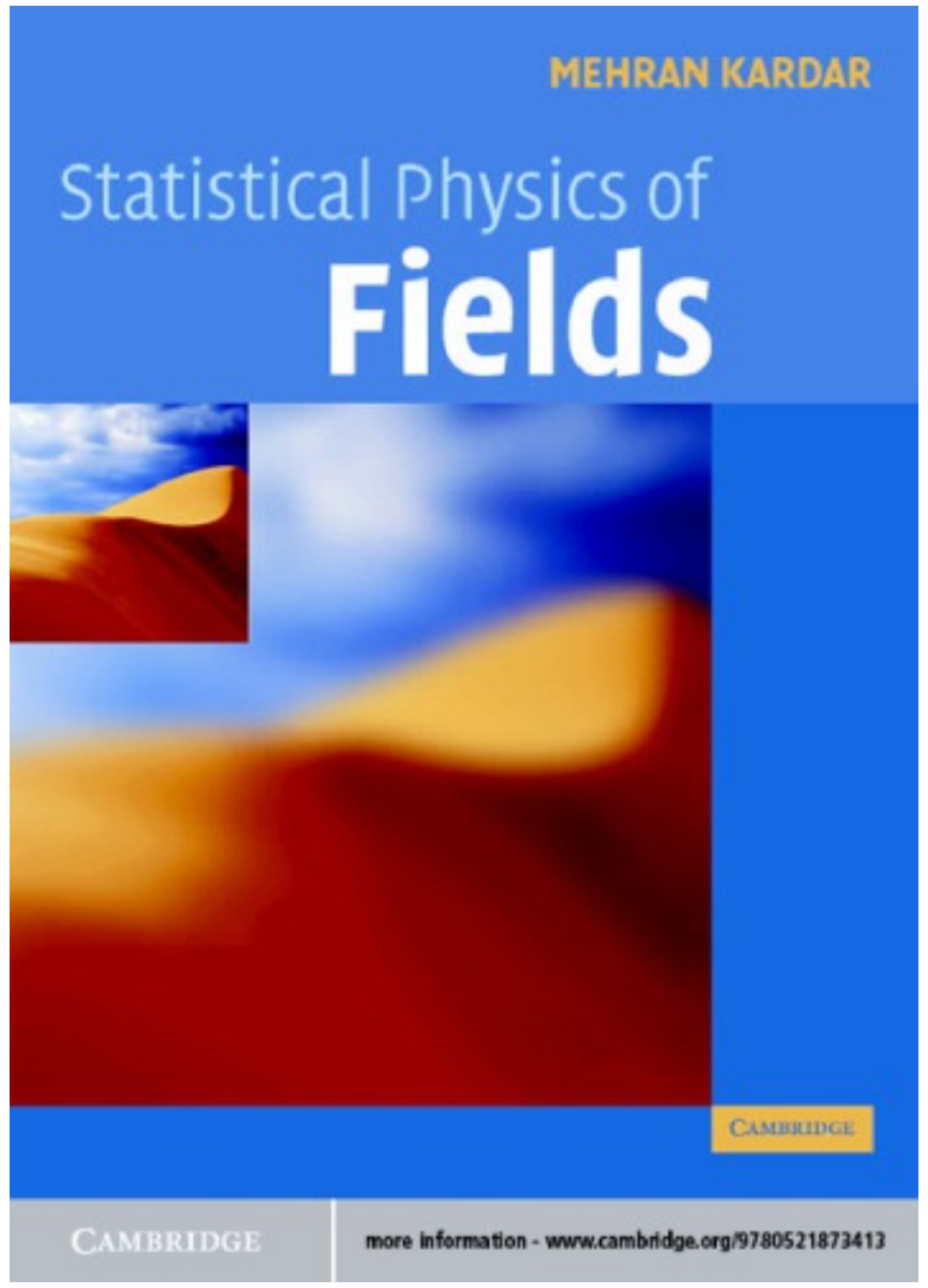
Keeping only terms at order of ϵ, we obtain

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

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

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

from Langevin to Fokker-Planck



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 (kind of layer-wise training)

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

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

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

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

Claim:

$$\mathbb{E}_{x\sim q(x)}|s_\theta(x)-\nabla_x \ln q(x)|^2 = \mathbb{E}_{x_0\sim q_0(x_0)}\mathbb{E}_{x\sim q(x|x_0)}|s_\theta(x)-\nabla_x \ln q(x|x_0)|^2 + \text{const}.$$

↓

Independent of θ

$$q(x) = \int q(x|x_0)q_0(x_0)dx_0$$

Proof:

$$\mathbb{E}_{x_0\sim q_0(x_0)}\mathbb{E}_{x\sim q(x|x_0)}|s|^2 = \int dx_0 \int dx q_0(x_0)q(x|x_0)|s|^2 = \int dx q(x)|s|^2 = \mathbb{E}_{x\sim q(x)}|s^2|$$

$$\mathbb{E}_{x_0\sim q_0(x_0)}\mathbb{E}_{x\sim q(x|x_0)}[s\cdot\nabla \ln q(x|x_0)] = \int dx_0 \int dx q_0(x_0)q(x|x_0)\frac{s\cdot\nabla q(x|x_0)}{q(x|x_0)}$$

$$= \int dx_0 \int dx q_0(x_0) s\cdot\nabla q(x|x_0)$$

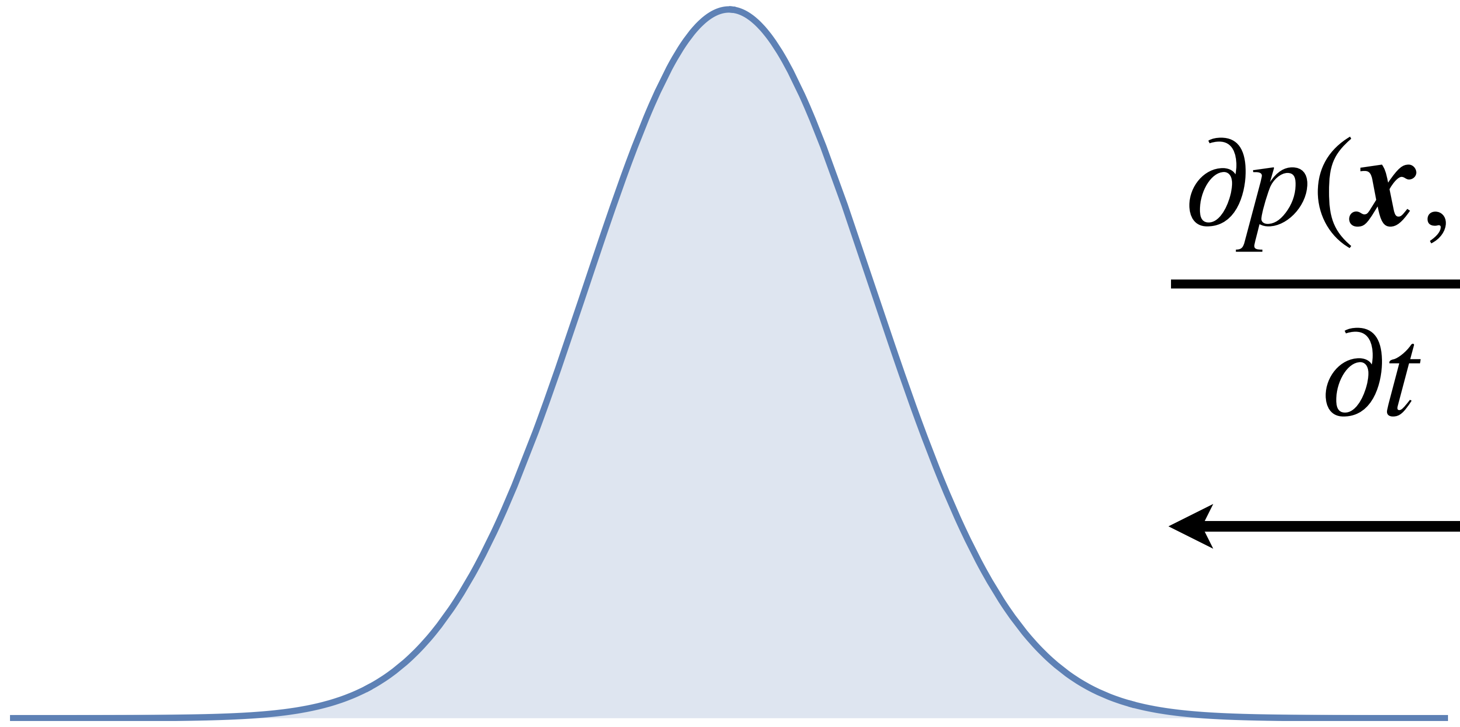
$$= \int dx s\cdot\nabla q(x) = \mathbb{E}_{x\sim q(x)}[s\cdot\nabla \ln q(x)]$$

要让$\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$等于$\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)$的加权平均【即$\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)$】只需要最小化$\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)-\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\|^2$的加权平均 https://spaces.ac.cn/archives/9209

Flow matching

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

base distribution



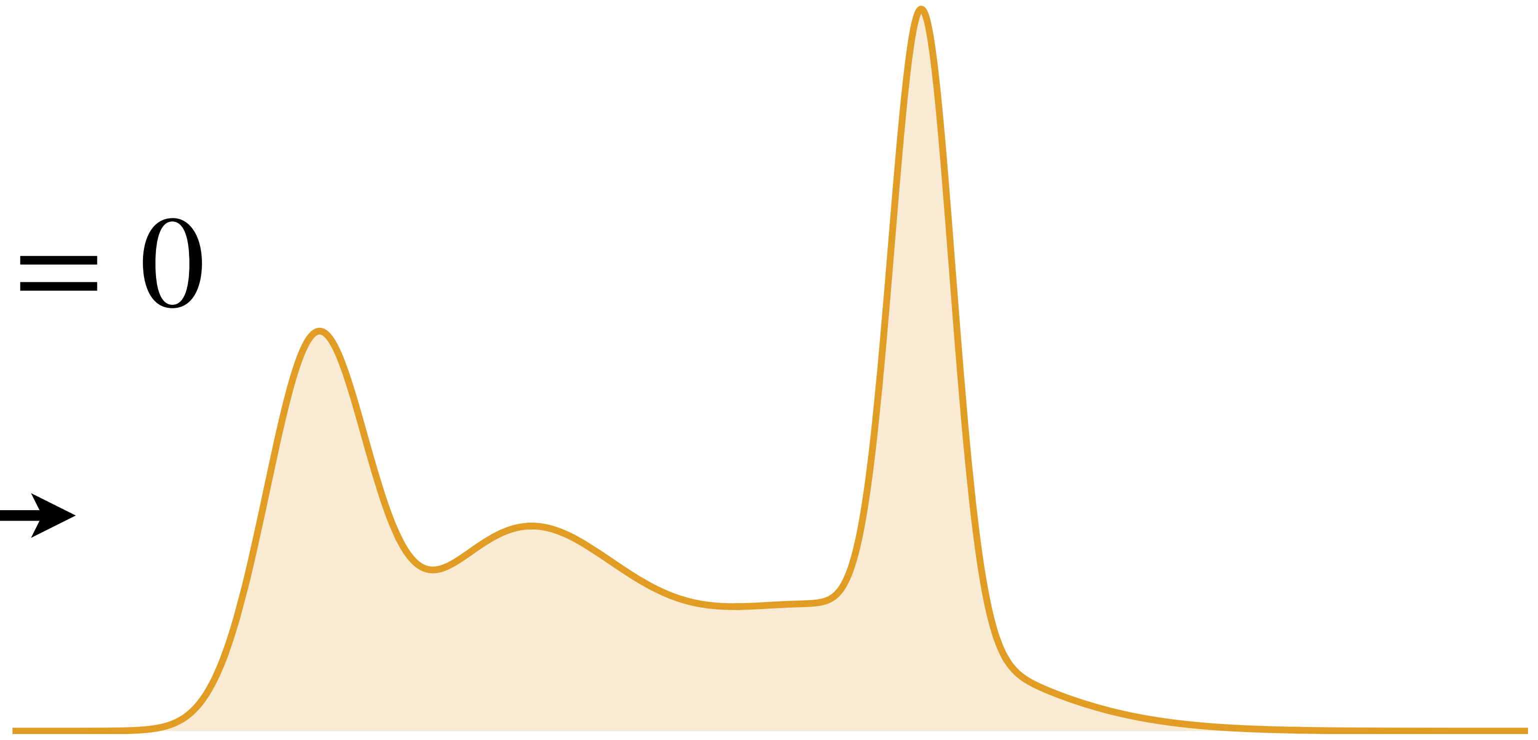
ground truth

velocity field

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

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

data distribution



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

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

Conditional flow matching

$$\frac{\partial p(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)\right] = 0$$

$$p(\boldsymbol{x}, t) = \int p(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)\, q(\boldsymbol{x}_1) d\boldsymbol{x}_1 \quad p(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t) = \int p(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)\, q(\boldsymbol{x}_1) d\boldsymbol{x}_1$$

$$\mathscr{L}_{\text{CFM}} = \mathbb{E}_{t\sim\mathscr{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_1\sim q(\boldsymbol{x}_1)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_1,t)} \left|\boldsymbol{v}_\theta(\boldsymbol{x}, t) - \boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t)\right|^2$$

$$\nabla_\theta \mathscr{L}_{\text{FM}} = \nabla_\theta \mathscr{L}_{\text{CFM}}$$

an example:

Rectified flow 2209.03003

causalizing linear interpolation

$$\boldsymbol{x} = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1 \qquad p(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t) = \mathscr{N}\left(t\boldsymbol{x}_1, (1-t)^2\right)$$

$$\boldsymbol{x}_0 \sim \mathscr{N}(0, I) \qquad \boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1, t) = d\boldsymbol{x}/dt = \boldsymbol{x}_1 - \boldsymbol{x}_0$$

Claim: $\nabla_\theta \mathscr{L}_{\mathrm{FM}} = \nabla_\theta \mathscr{L}_{\mathrm{CFM}}$

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

$$\mathscr{L}_{\mathrm{CFM}} = \mathbb{E}_{t\sim\mathscr{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_1\sim q(\boldsymbol{x}_1)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_1,t)} \left|\boldsymbol{v}_\theta(\boldsymbol{x},t) - \boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\right|^2$$

$$p(\boldsymbol{x},t) = \int p(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\,q(\boldsymbol{x}_1)d\boldsymbol{x}_1 \qquad p(\boldsymbol{x},t)\boldsymbol{u}(\boldsymbol{x},t) = \int p(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\,q(\boldsymbol{x}_1)d\boldsymbol{x}_1$$

Proof:

$$\mathbb{E}_{\boldsymbol{x}_1\sim q(\boldsymbol{x}_1)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_1,t)} \left|\boldsymbol{v}_\theta\right|^2 = \int d\boldsymbol{x}_1 \int d\boldsymbol{x} q(\boldsymbol{x}_1)p(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\,|\boldsymbol{v}_\theta|^2 = \int d\boldsymbol{x} p(\boldsymbol{x},t)\,|\boldsymbol{v}_\theta|^2 = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x},t)} \left|\boldsymbol{v}_\theta\right|^2$$

$$\mathbb{E}_{\boldsymbol{x}_1\sim q(\boldsymbol{x}_1)}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{x}_1,t)} \left[\boldsymbol{v}_\theta\cdot\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\right] = \int d\boldsymbol{x}_1 \int d\boldsymbol{x} q(\boldsymbol{x}_1)p(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\left[\boldsymbol{v}_\theta\cdot\boldsymbol{u}(\boldsymbol{x}\,|\,\boldsymbol{x}_1,t)\right]$$

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

Flow matching is all you need!

This framework contains various diffusion models as special cases

Optimal transport theory and iterative improvement of the interpolation path (Liu et al 2209.03003)

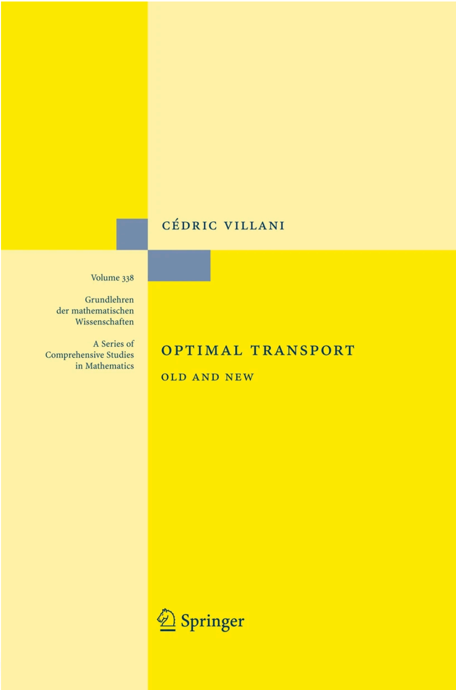
400x speedup compared to continuous normalizing flow (Albergo et al, 2209.15571)

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

Fun to try: flow matching for computing free energy difference

Fun to try: Train Riemannian flows with it

Part II Optimal transport and Riemannian geometry

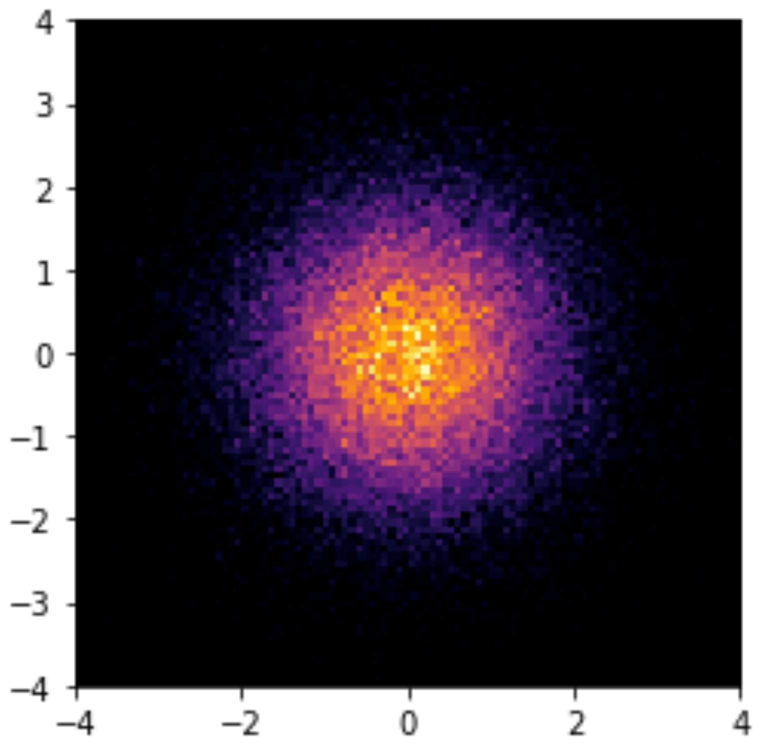


Demo: Classical Coulomb gas in a harmonic trap

Estimating free energy via flow matching

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

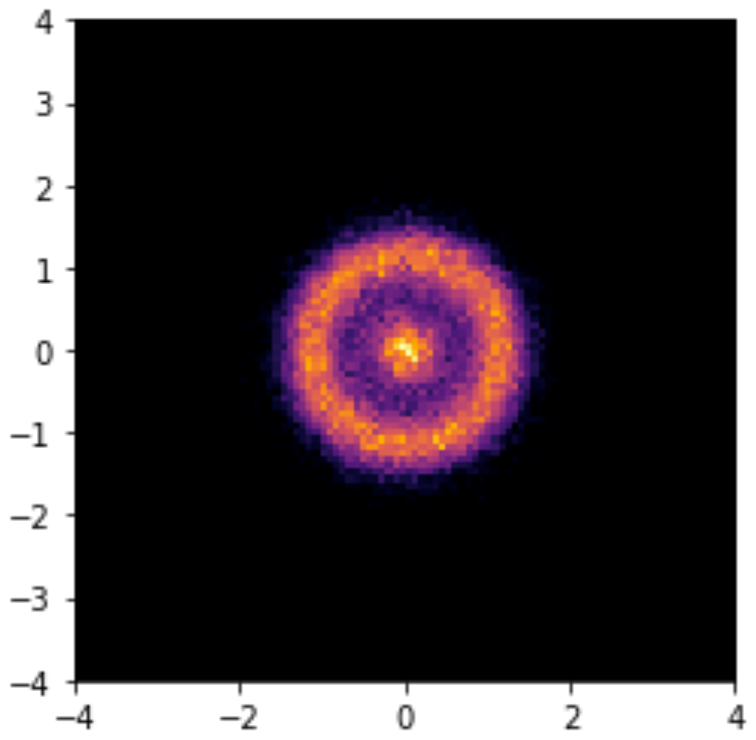
$$Z = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[e^{-\beta E(\boldsymbol{x})-\ln q(\boldsymbol{x})}\right] \qquad \ln q(\boldsymbol{x}) = \ln\mathcal{N}(0,I) - \int_0^1 \nabla\cdot\boldsymbol{v}dt$$



Base density

direct sampling

Interpolate



Target density

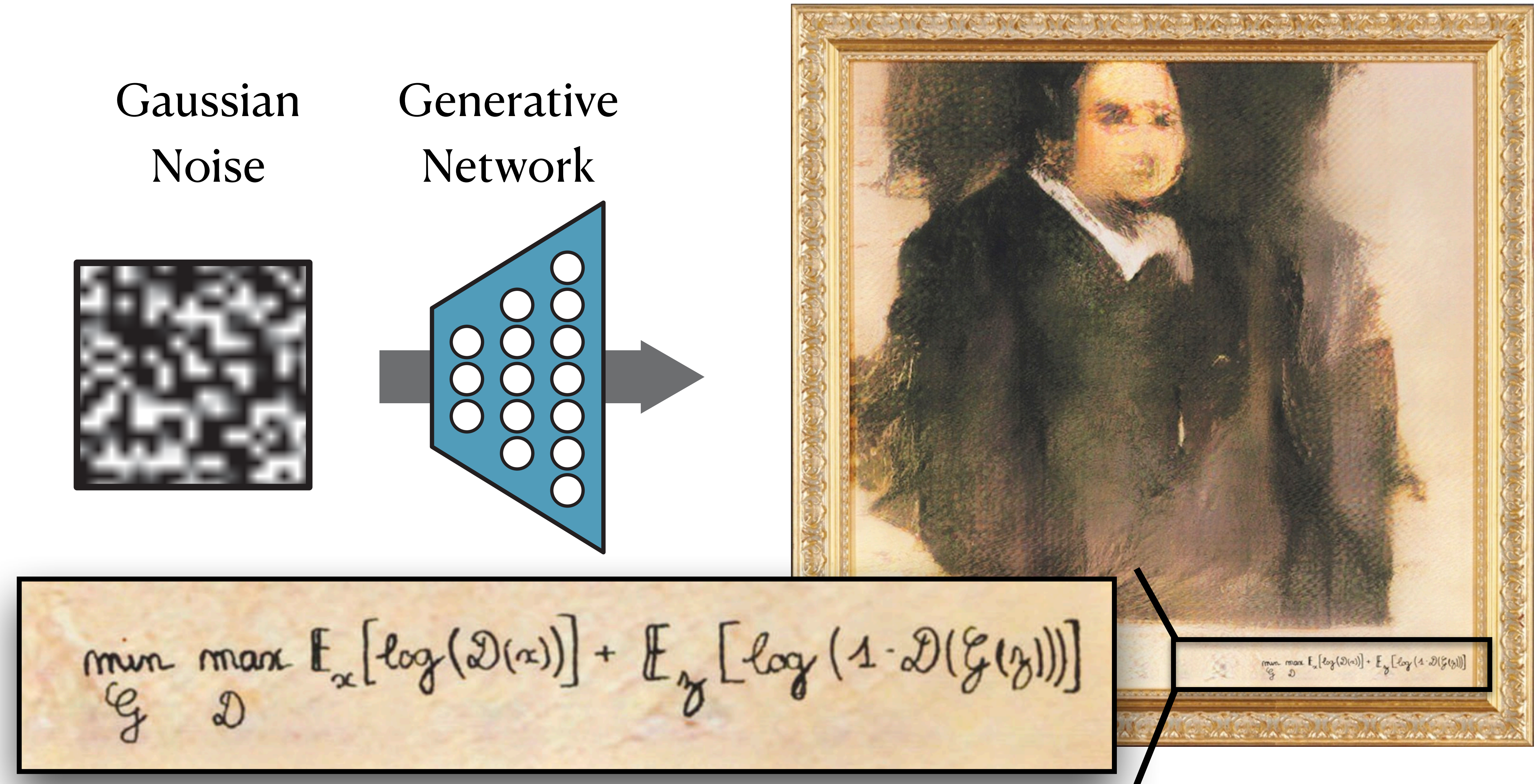
MCMC sampling

GAN

Likelihood free simulator

Prone to mode collapse

More tricky to train than others



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

Performance have been surpassed by diffusion models

I found GAN to be less useful for serious scientific applications

VAE

Close connection to variational calculus we have just learned

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

Variational free energy

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

Variational Bayes

2209.10423

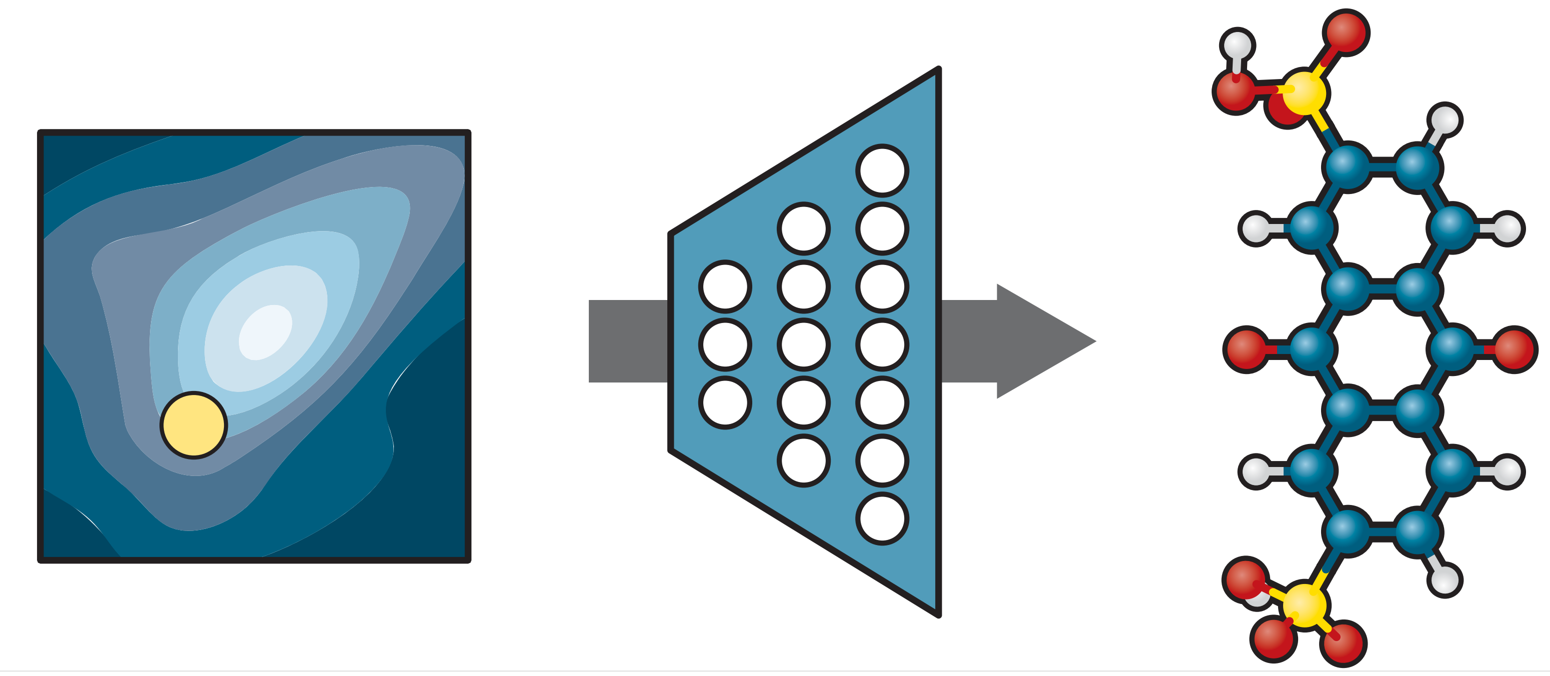
Approximate sampling and estimation of partition functions using neural networks

George T. Cantwell
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM, 87501
gcant@umich.edu

We consider the closely related problems of sampling from a distribution known up to a normalizing constant, and estimating said normalizing constant. We show how variational autoencoders (VAEs) can be applied to this task. In their standard applications, VAEs are trained to fit data drawn from an unknown and intractable distribution. We invert the logic and train the VAE to fit a simple and tractable distribution, on the assumption of a complex and intractable latent distribution, specified up to normalization. This procedure constructs approximations without the use of training data or Markov chain Monte Carlo sampling. We illustrate our method on three examples: the Ising model, graph clustering, and ranking.

Generative AI for Science

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

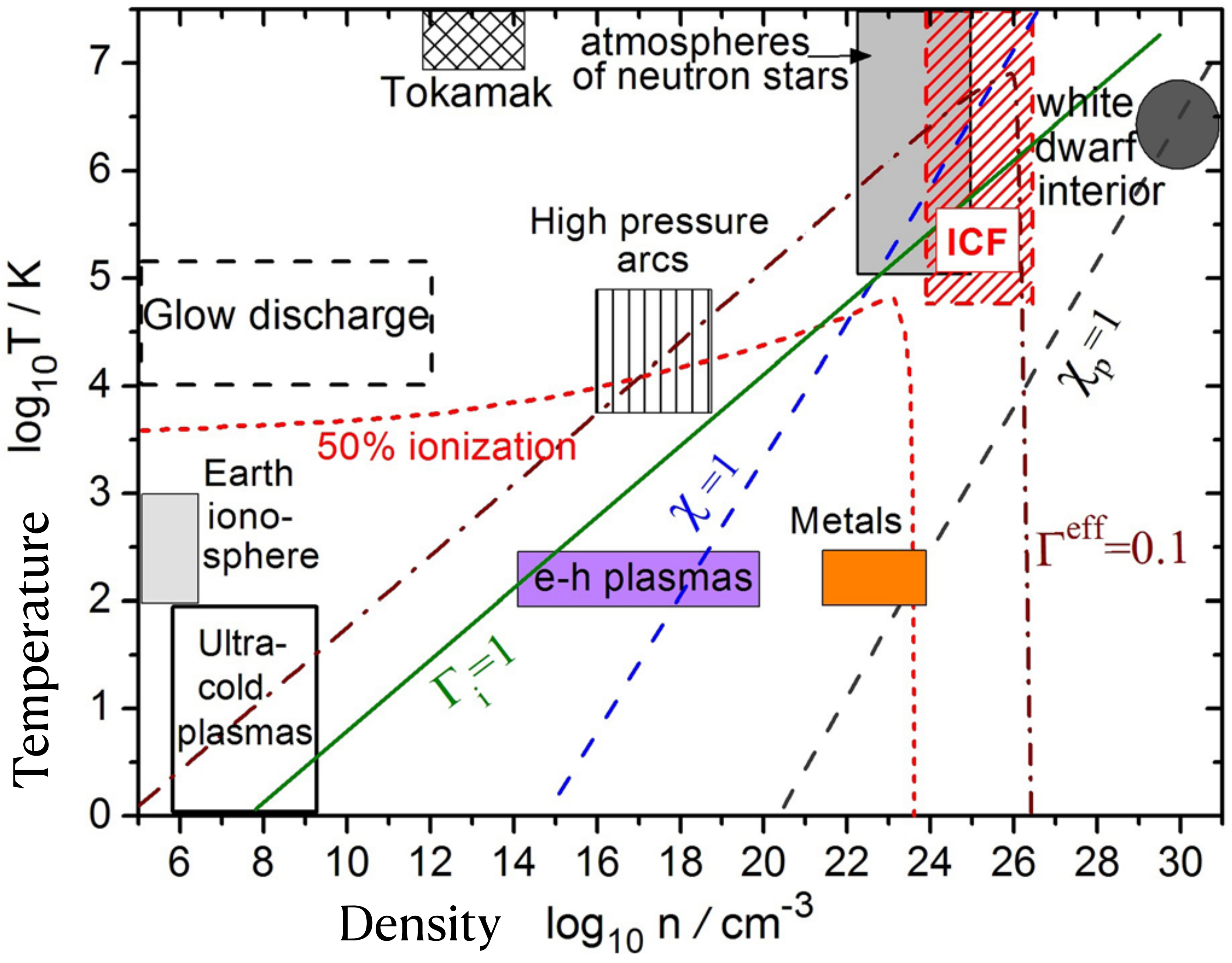
2.  Matter inverse design

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

Ab-initio study of quantum matters at T>0

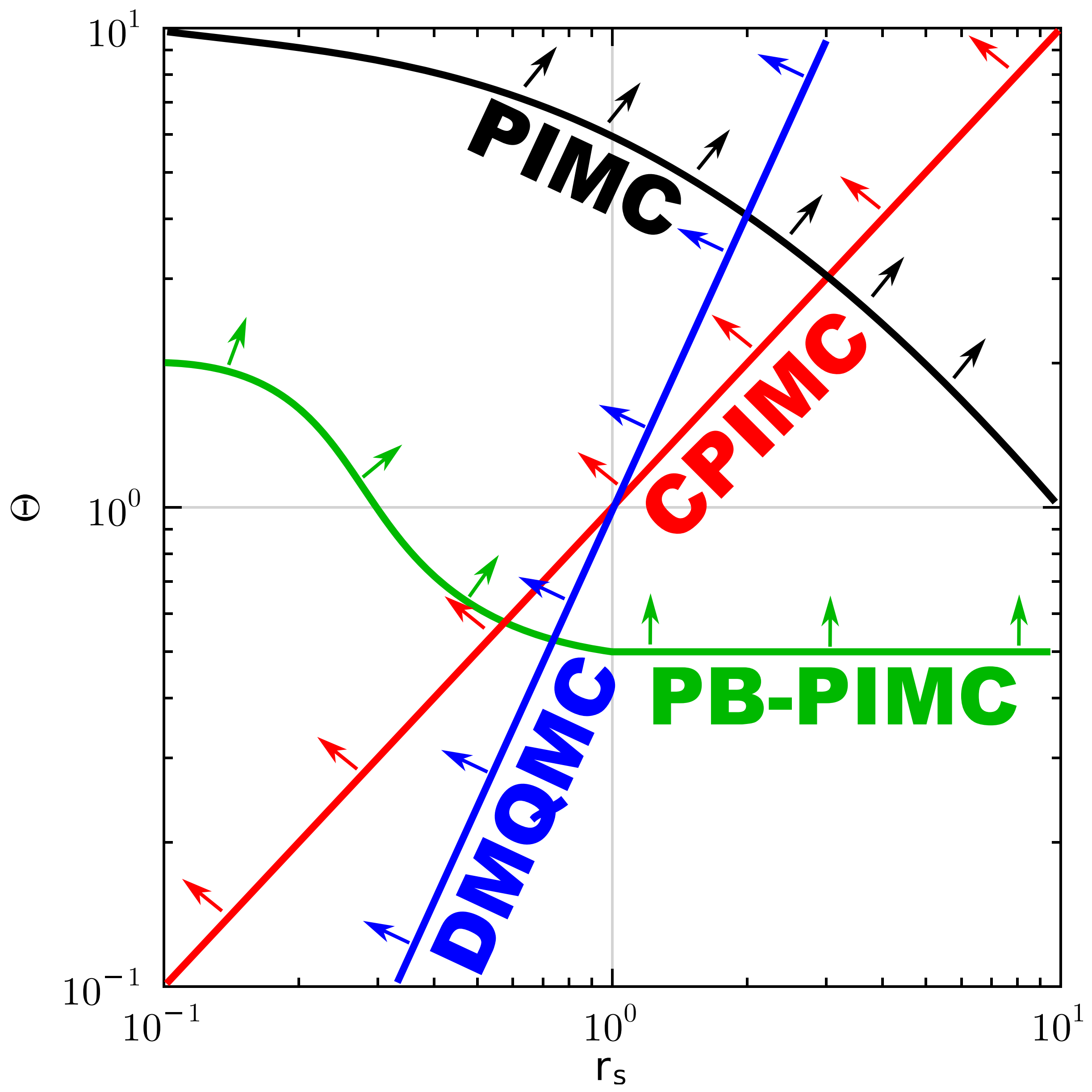
$$H = -\sum_i \frac{\hbar^2}{2m_e}\nabla_i^2 - \sum_I \frac{\hbar^2}{2m_I}\nabla_I^2 - \sum_{I,i}\frac{Z_I e^2}{|R_I - r_i|} + \frac{1}{2}\sum_{i\neq j}\frac{e^2}{|r_i - r_j|} + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J e^2}{|R_I - R_J|}$$

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



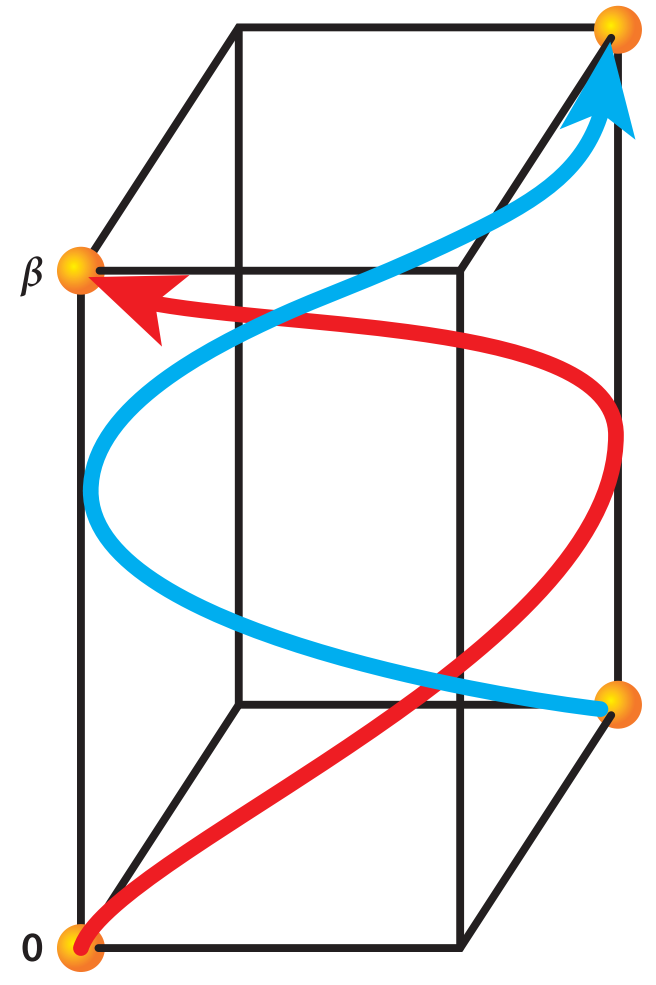
Bonitz et al, Phys. Plasmas '20

Application range
of quantum Monte Carlo



Dornheim et al, Phys. Plasmas '17

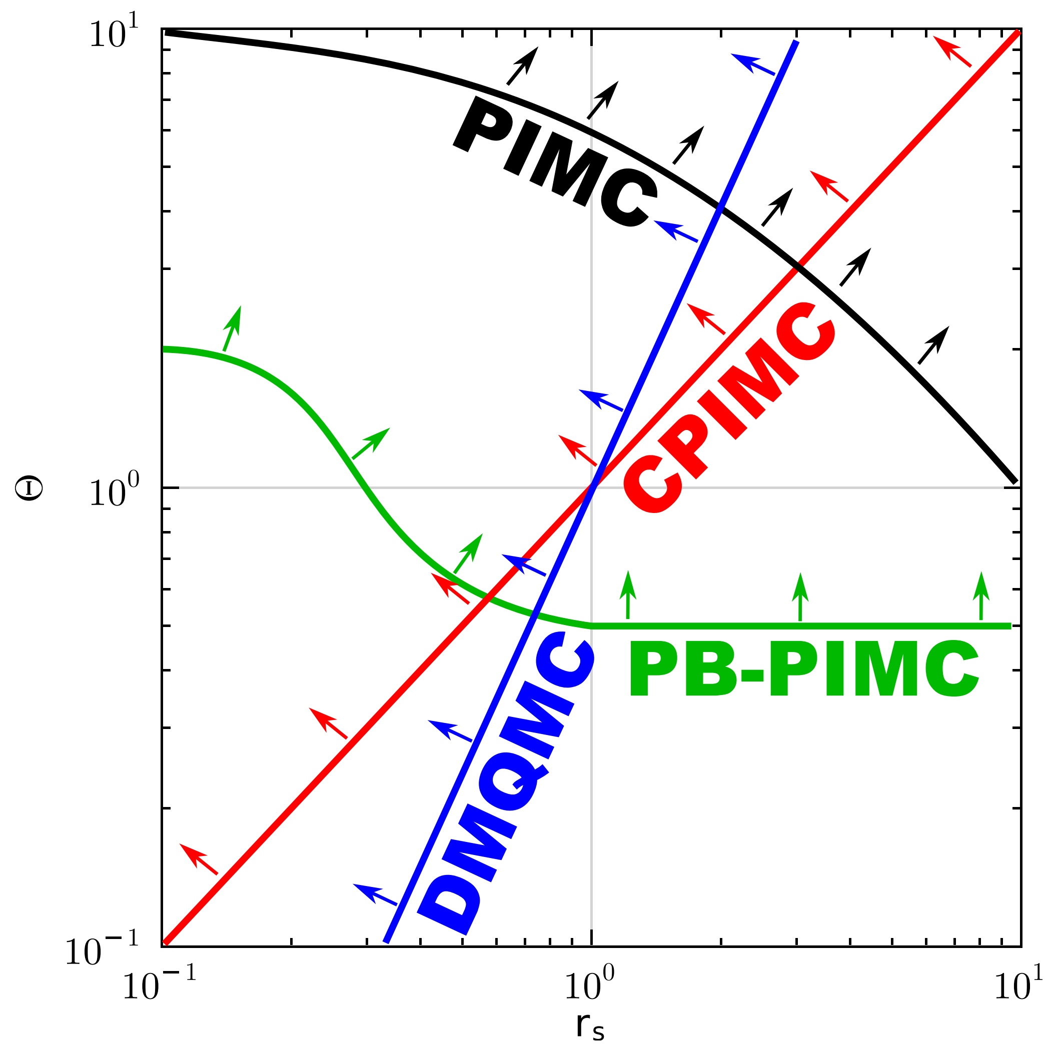
How to solve quantum many-body systems?



Quantum-to-classical mapping via path integral, then we are done

$$Z = \int d\tau \, d\boldsymbol{x} \cdots$$

However, the sign problem strikes again: the "weights" may not be positive 😂



We need a variational principle that directly applies to quantum systems

The Gibbs-Feynman-Bogolyubov-Delbrück–Molière variational principle

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

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

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

Richard P. Feynman

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

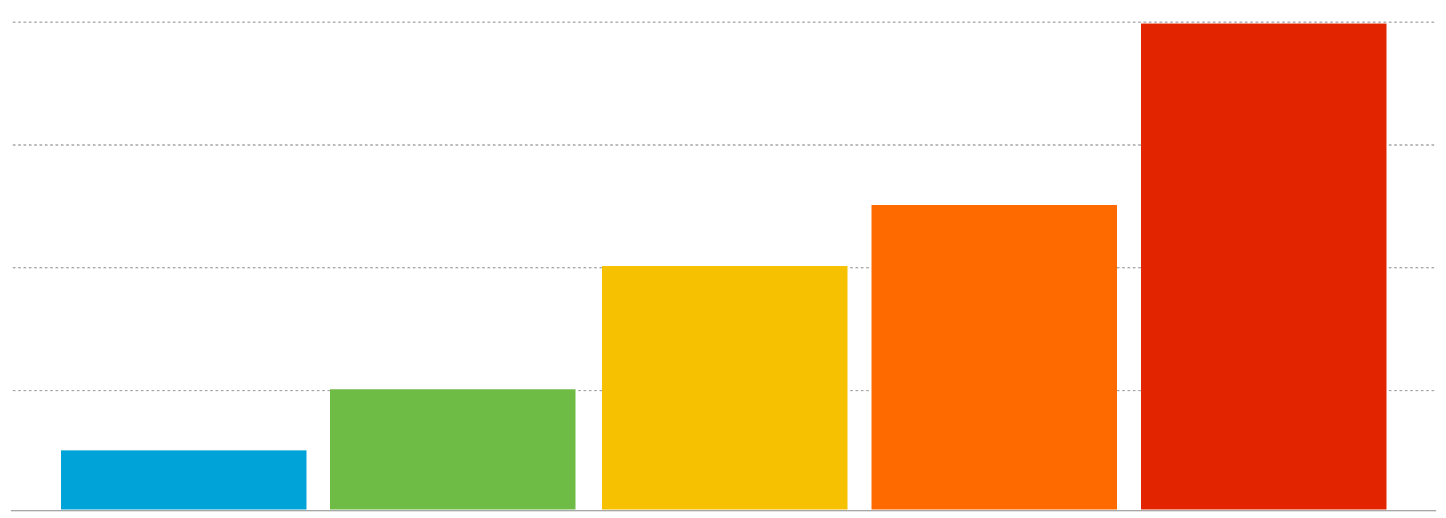
deep generative models !!

Classical world	Quantum world
Probability density p	Density matrix ρ
Kullback-Leibler divergence $\mathbb{KL}\left(p\,||\,q\right)$	Quantum relative entropy $S\left(\rho\,||\,\sigma\right)$
Variational free-energy	Variational free-energy
$F = \int d\boldsymbol{x}\ \left[\frac{1}{\beta} p(\boldsymbol{x}) \ln p(\boldsymbol{x}) + p(\boldsymbol{x}) H(\boldsymbol{x})\right]$	$F = \frac{1}{\beta}\mathrm{Tr}(\rho \ln \rho) + \mathrm{Tr}(\rho H)$

Density matrix

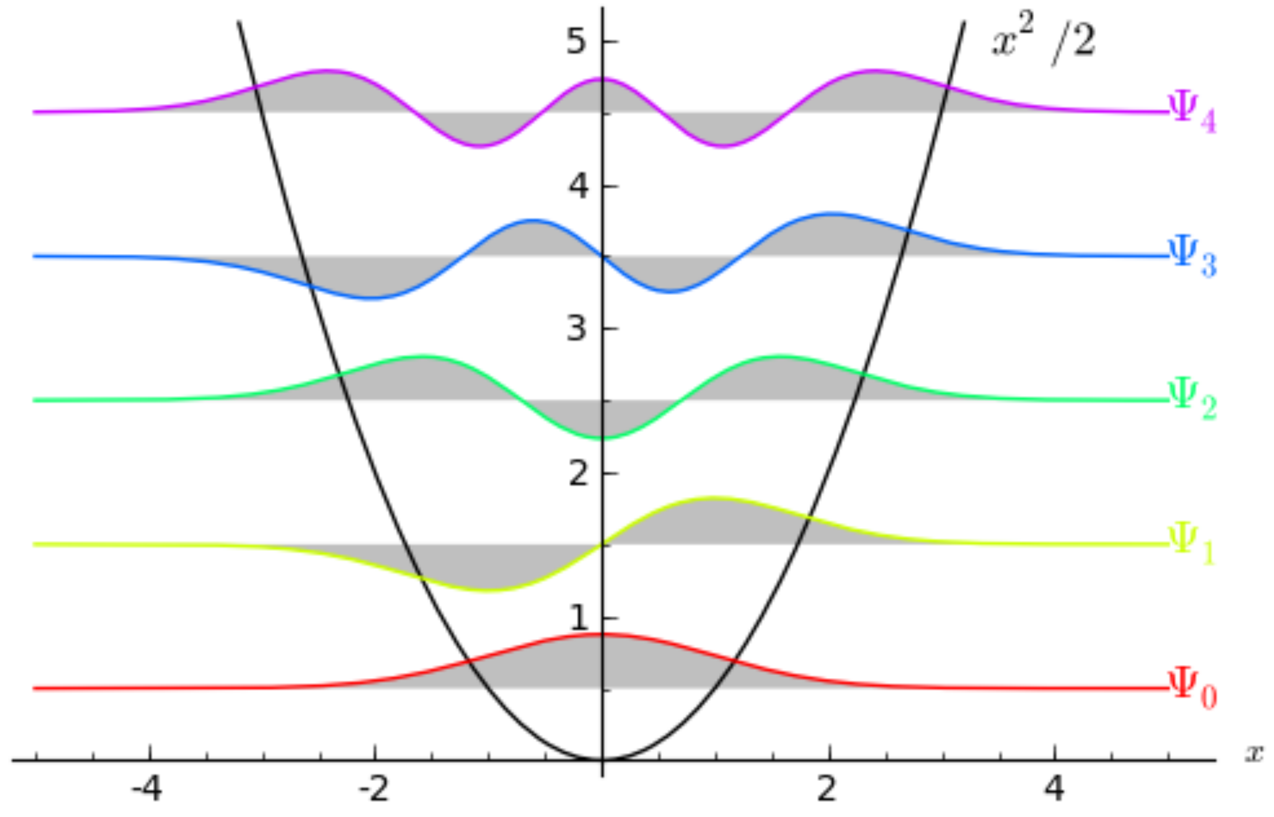
$$\rho = \sum_n p_n |\Psi_n\rangle\langle\Psi_n|$$

Classical probability $0 < p_n < 1$



$$\sum_n p_n = 1$$

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



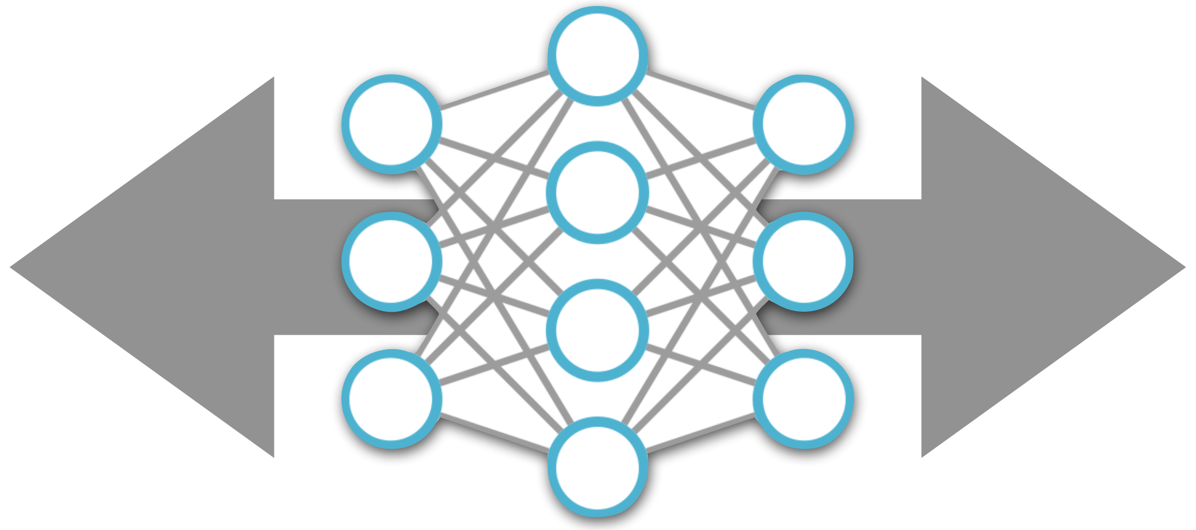
$$\langle \Psi_m | \Psi_n \rangle = \delta_{mn}$$

How to represent them ??

Use TWO deep generative models !!

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

Particle coordinates x



z Quasi-particle coordinates

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

Target states

Base states

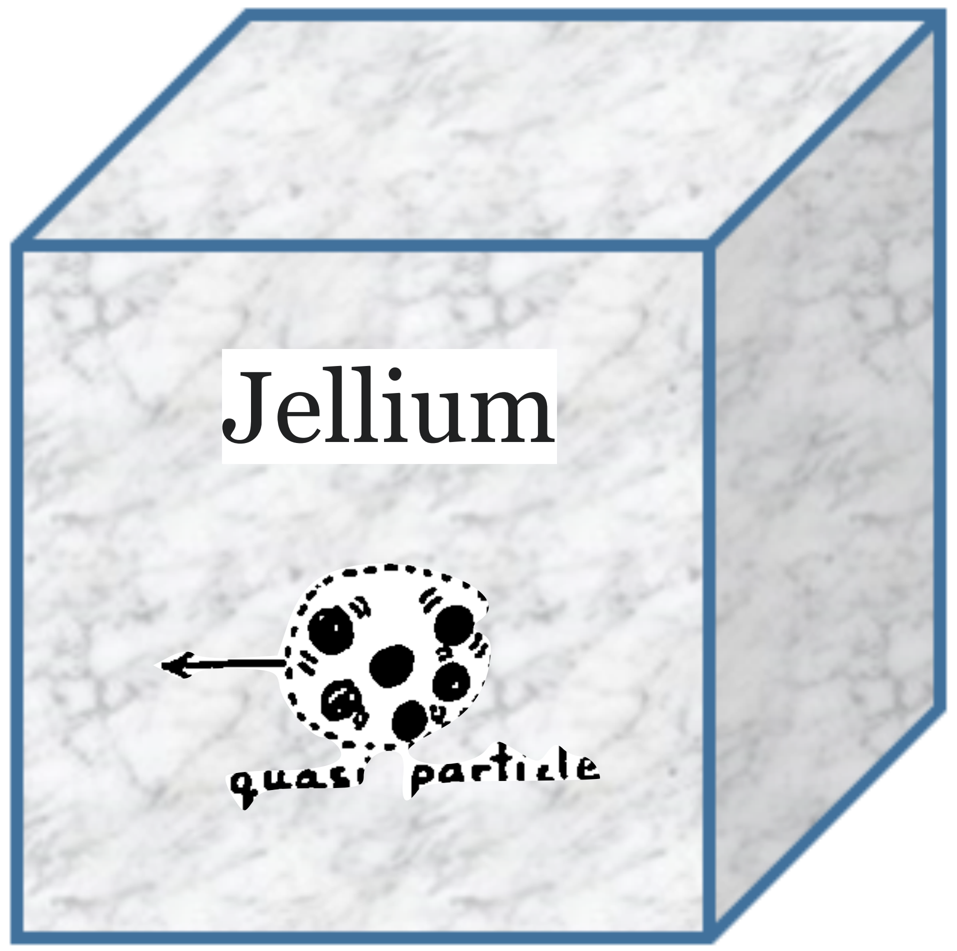
Jacobian of the flow

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

Applications to two prototypical quantum many-body problems

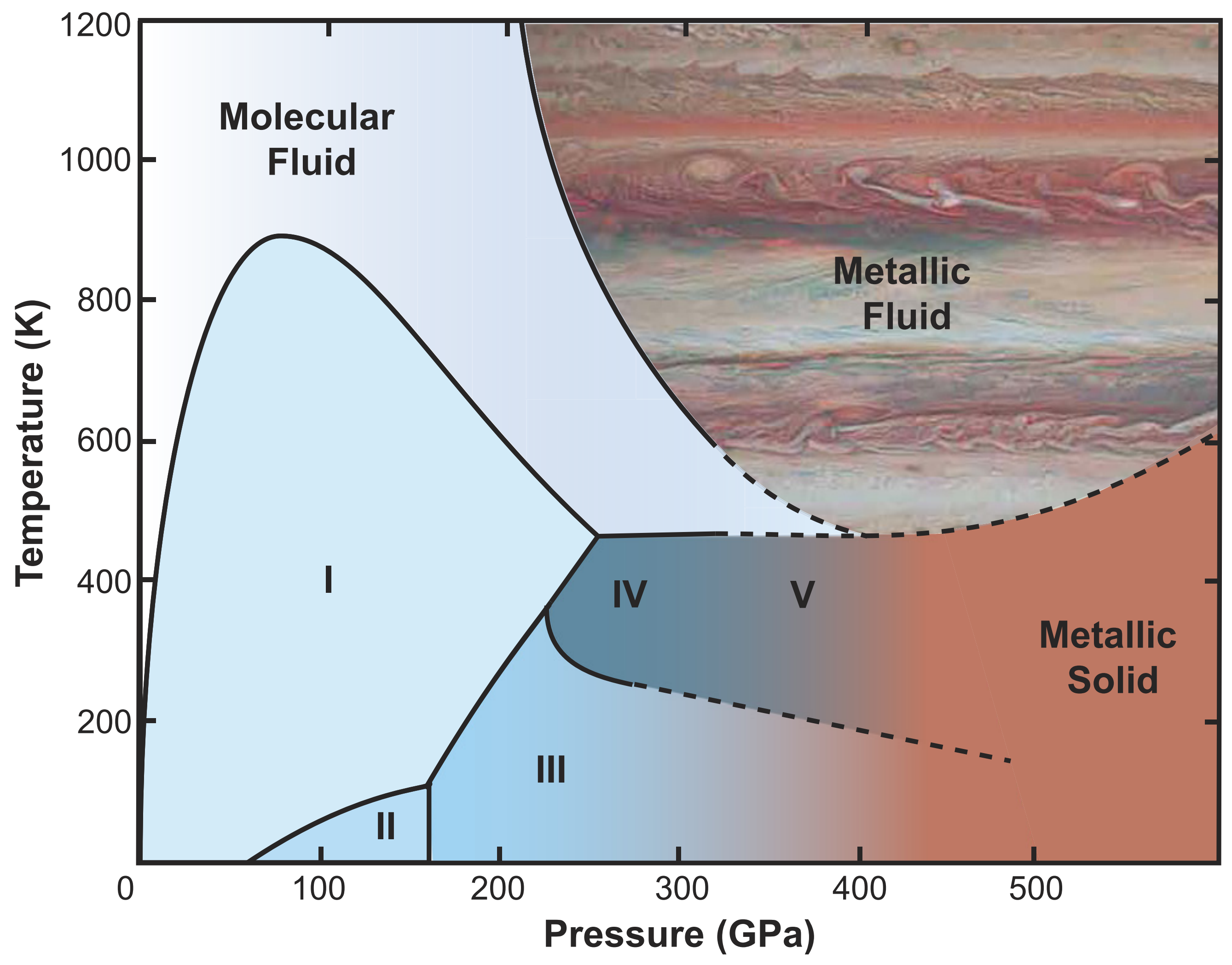
Uniform electron gas



Xie, Zhang, LW, 2201.03156

Dense hydrogen

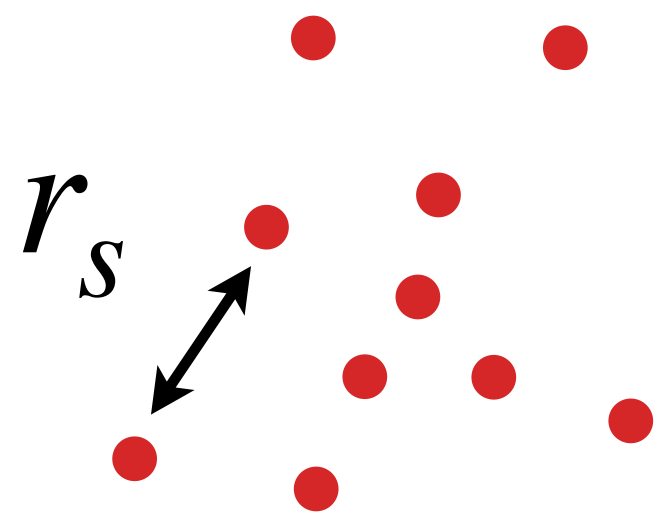
Gregoryanz et al, Matter Radiat. Extremes, 2020



Xie, Li, Wang, Zhang, LW, 2209.06095

Uniform electron gas

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



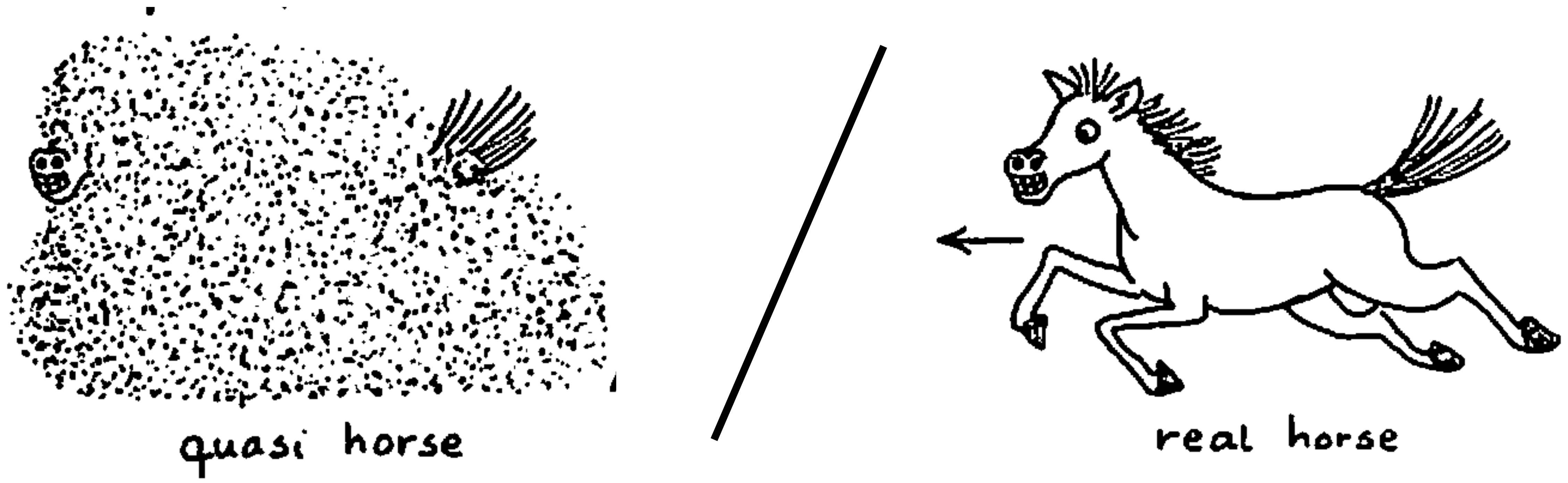
Fundamental model
for metals ($2 < r_s < 6$)

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

Input to the density
functional theory calculations

Quasi-particles effective mass

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



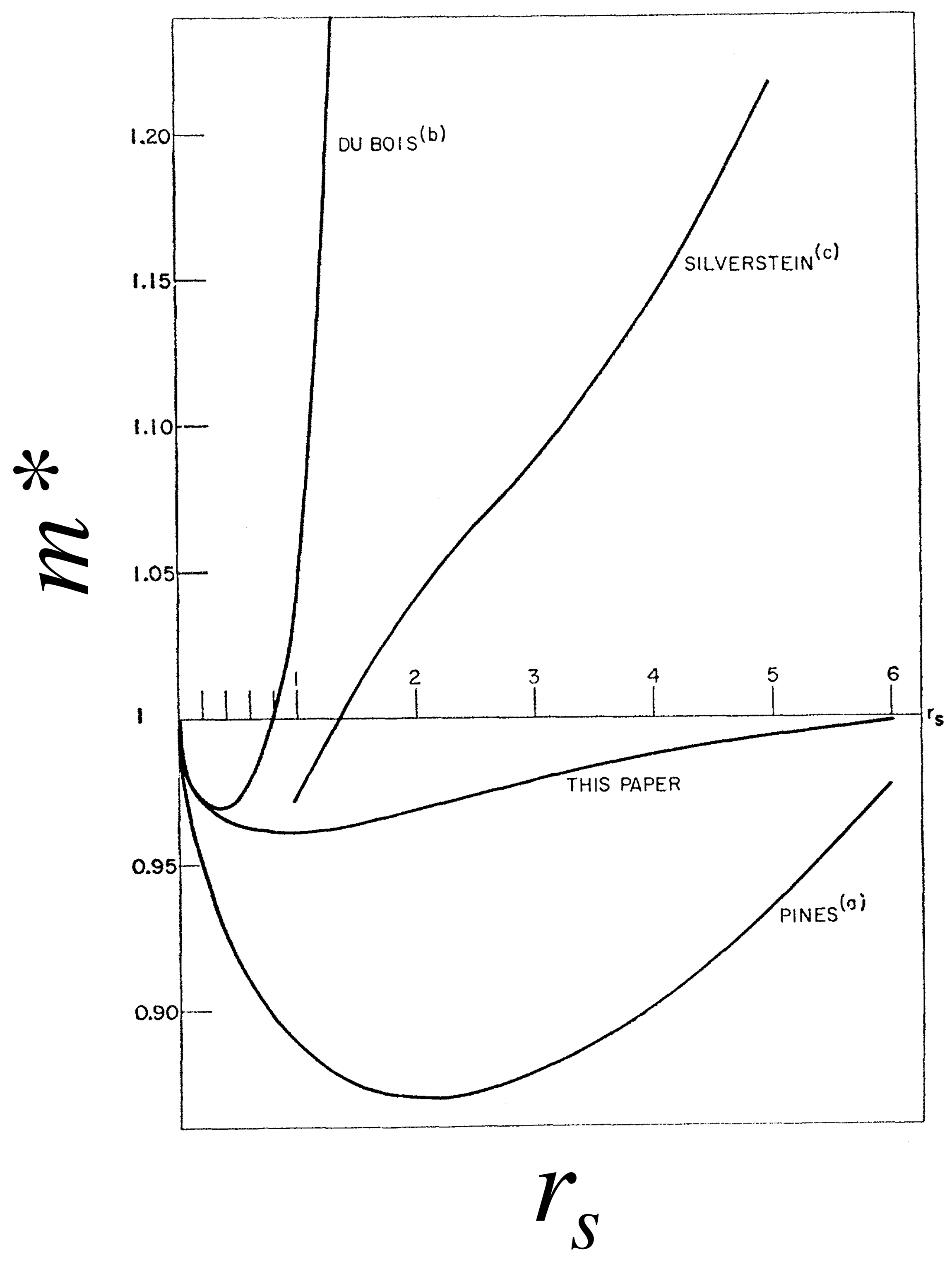
Richard D. Mattuck
A Guide to Feynman Diagrams in the Many-body Problem

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

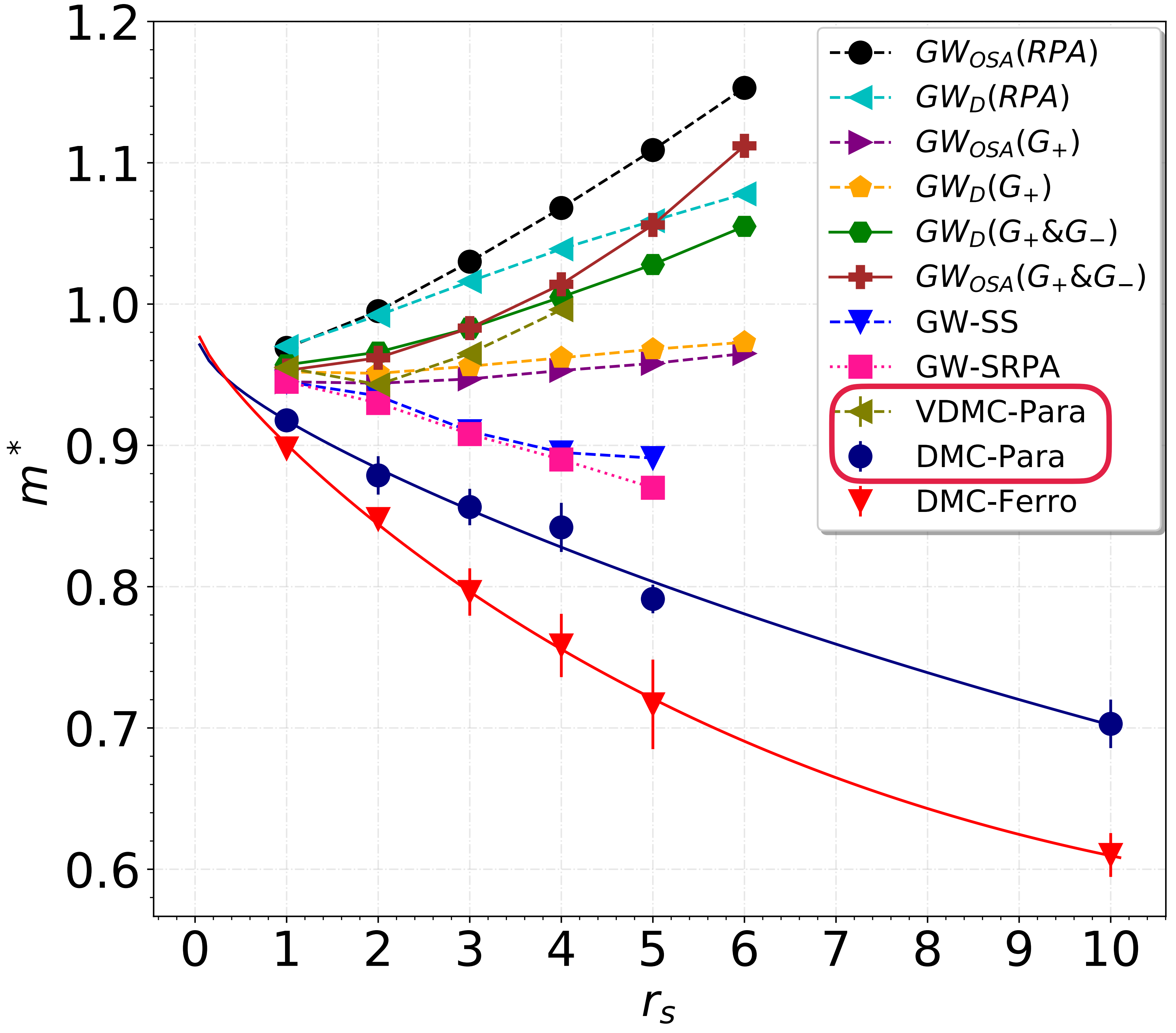
$N(0)$	s	c_V	χ
Density of states	entropy	specific heat	magnetic susceptibility

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

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

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

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

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

s ← interacting electrons

s_0 ← noninteracting electrons

Not an easy task due to the lack of reliable methods
for interacting electrons at low-temperature with intermediate density

Deep generative models for the variational density matrix

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

Normalized probability distribution

Orthonormal many-electron basis

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

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

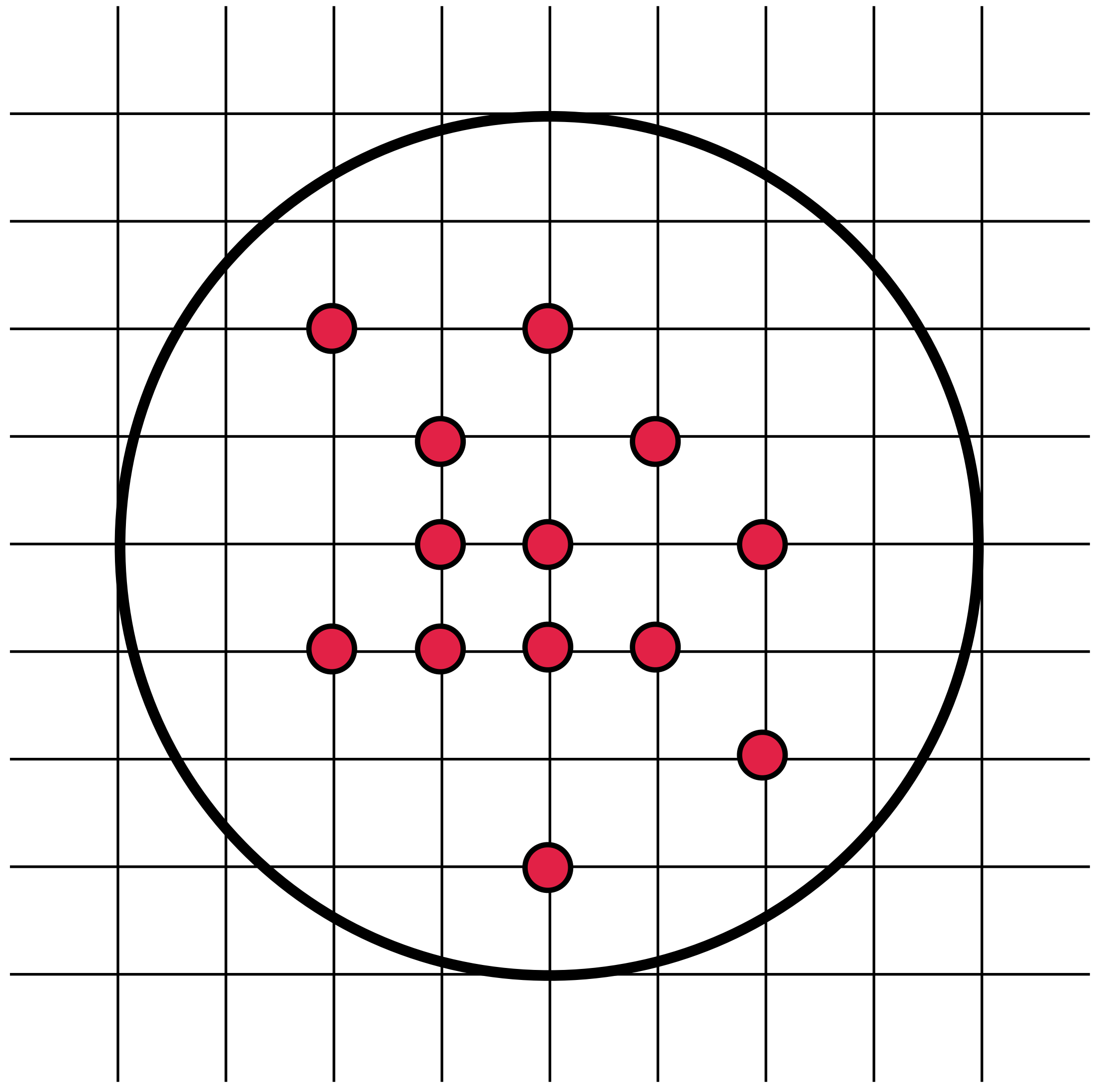
There will also be interesting twists for physics considerations

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

Fermionic occupation in k-space

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

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

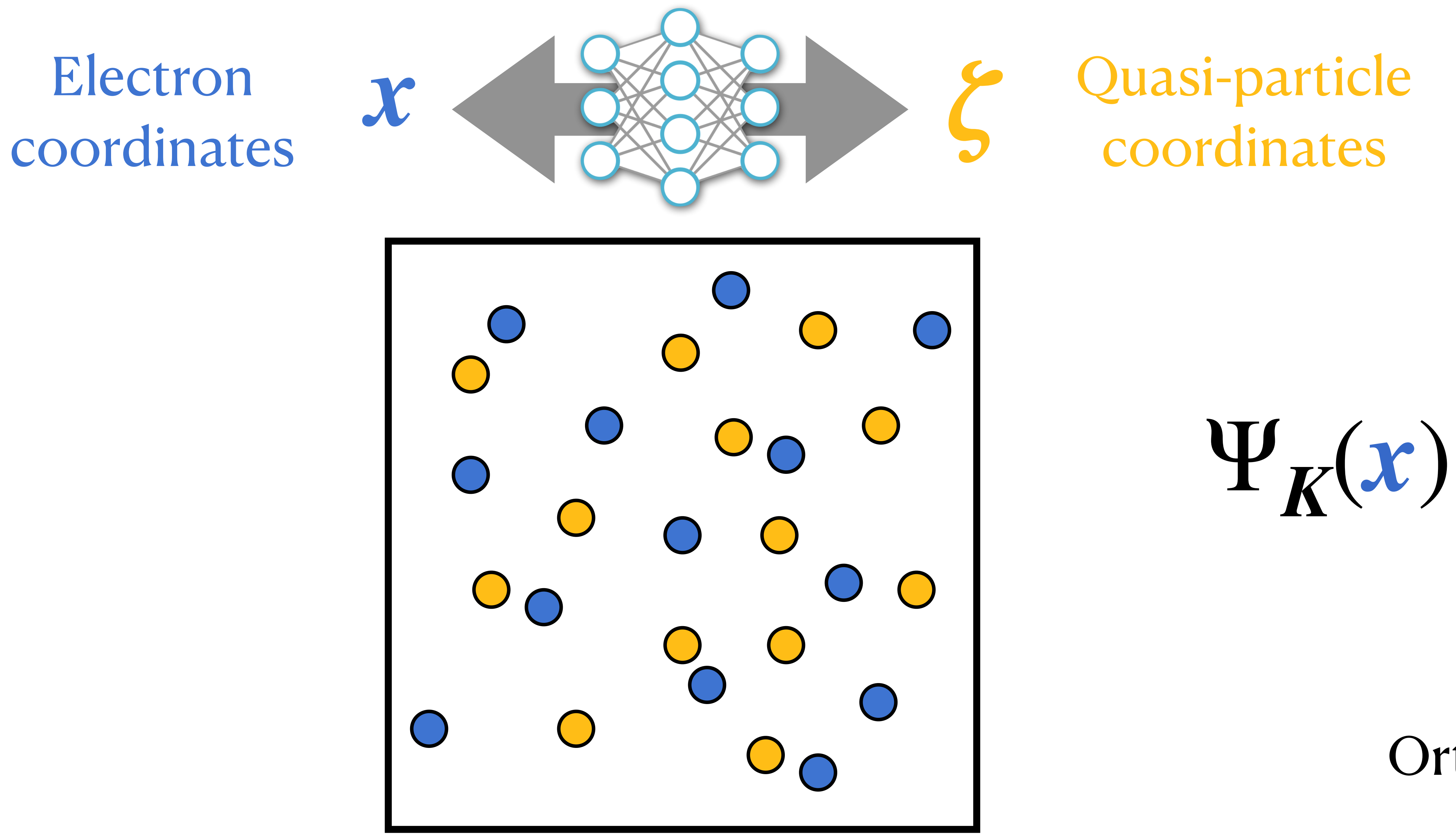


# of fermions	# of words
Momentum cutoff	Vocabulary
Entropy	Negative log-likelihood

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

❷ $\sqrt{\text{Normalizing flow for}}\ |\Psi_{\boldsymbol{K}}\rangle$



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

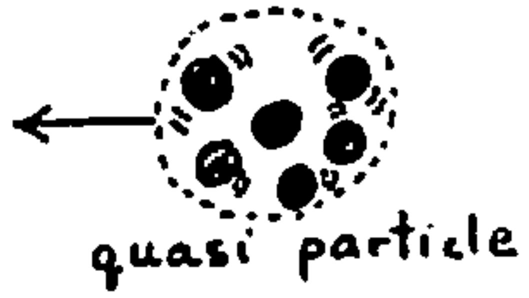
Orthonormal many-body states

Jacobian of the transformation

Twist: the flow should be permutation equivariant for fermionic coordinates

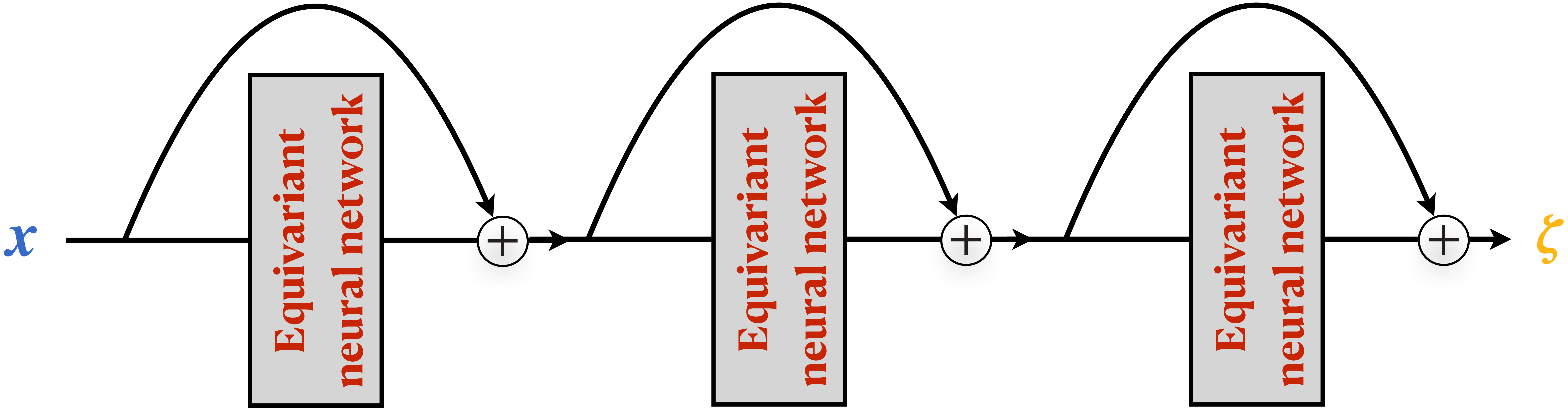
we use FermiNet layer Pfau et al, 1909.02487

Feynman's backflow in the deep learning era



$$\zeta_i = \boldsymbol{x}_i + \sum_{j \neq i} \eta(|\boldsymbol{x}_i - \boldsymbol{x}_j|)(\boldsymbol{x}_j - \boldsymbol{x}_i)$$

Feynman & Cohen 1956
wavefunction for liquid Helium



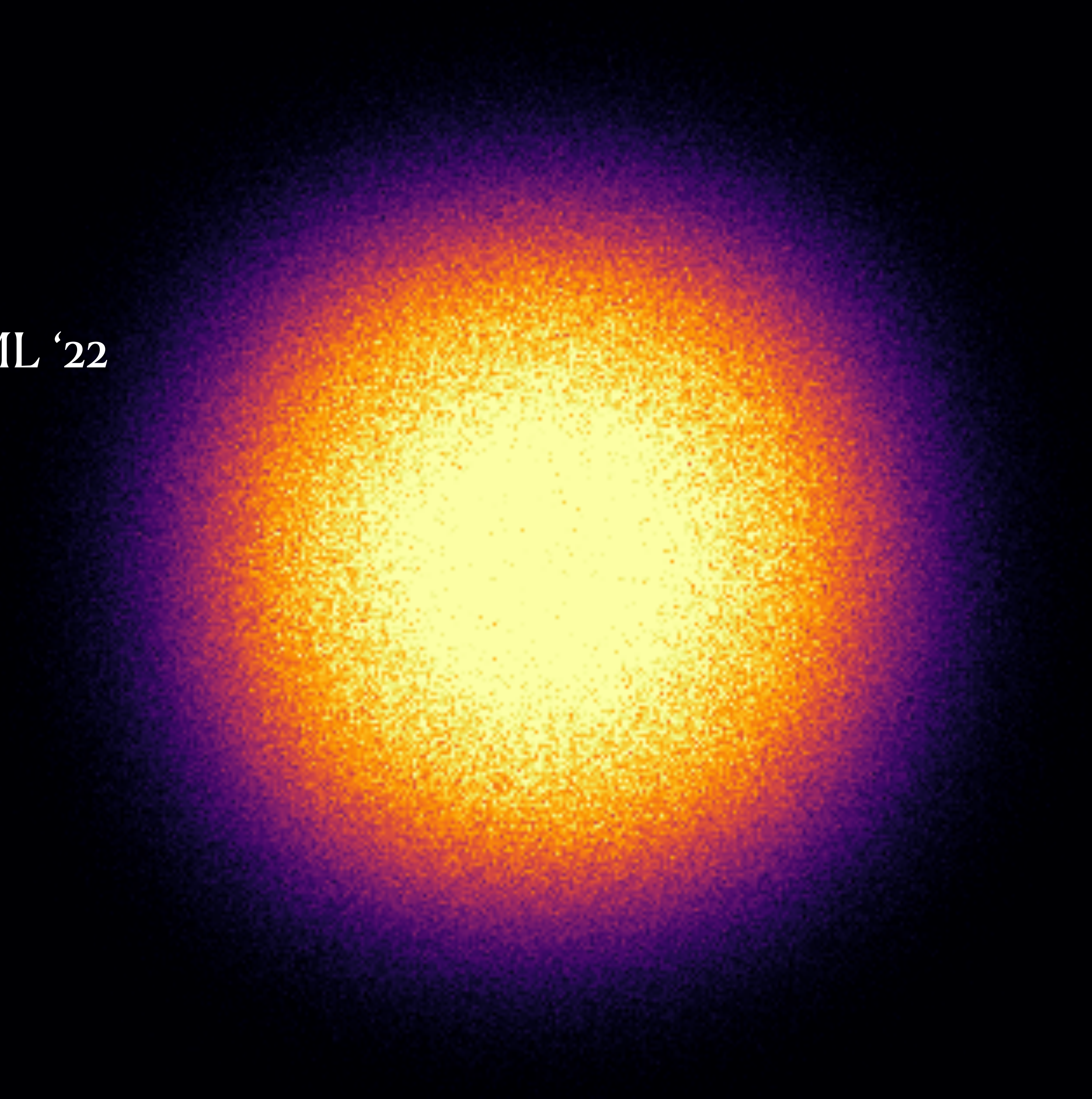
Twist: Iterative backflow → deep residual network → continuous normalizing flow

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

The objective function

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

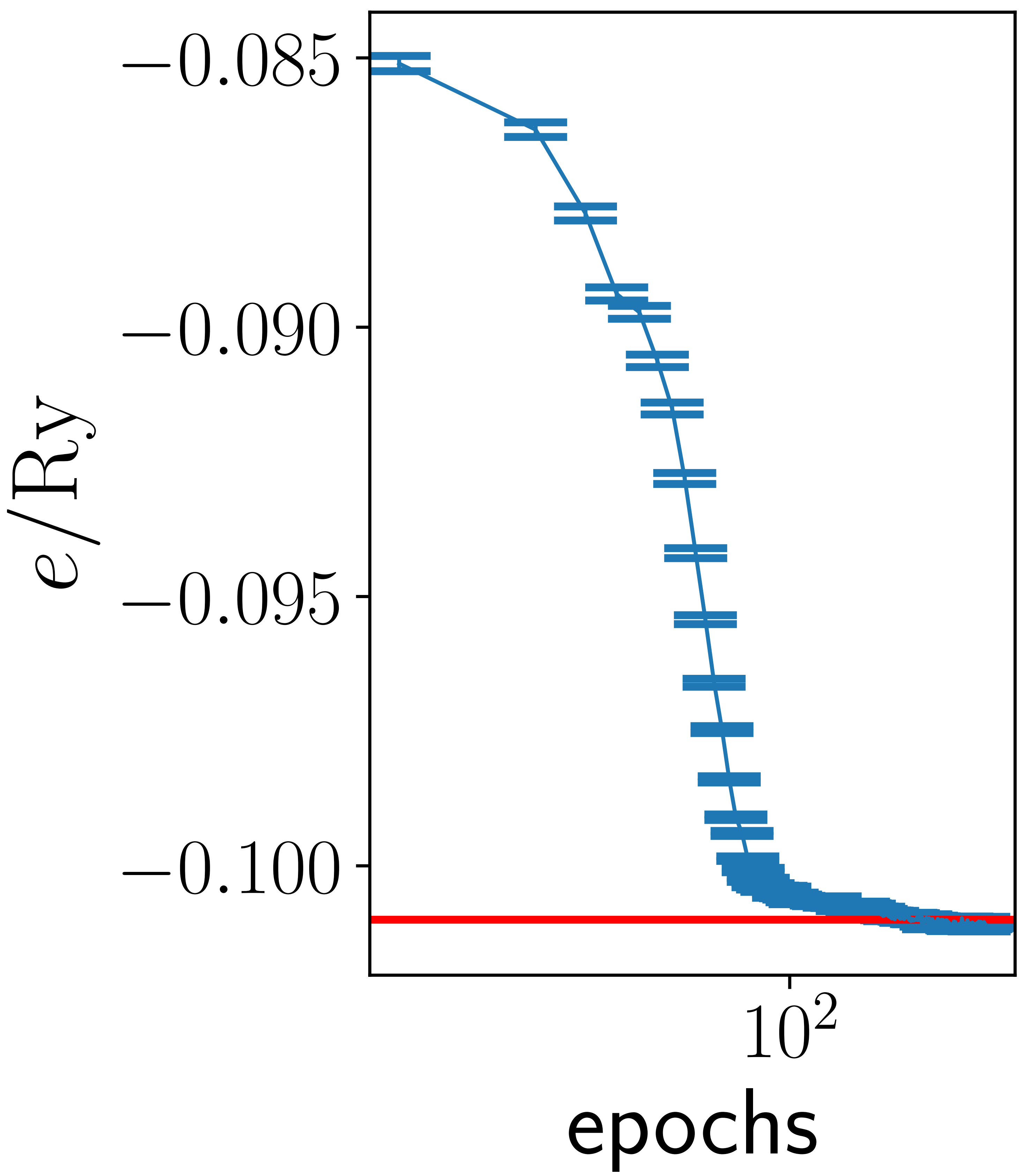
Boltzmann distribution

Born probability

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

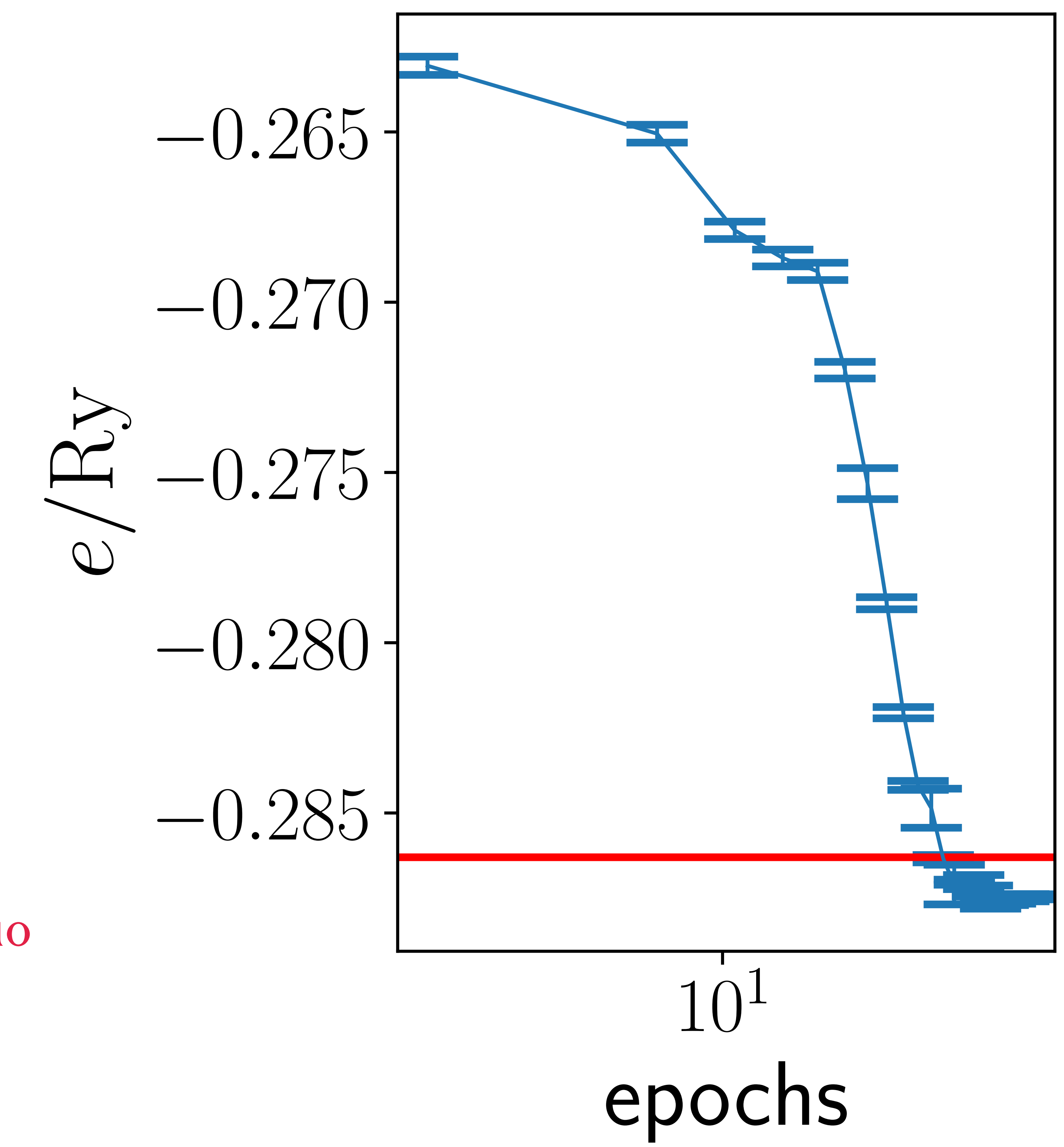
Benchmarks on spin-polarized electron gases

3D electron gas T/T_F=0.0625



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

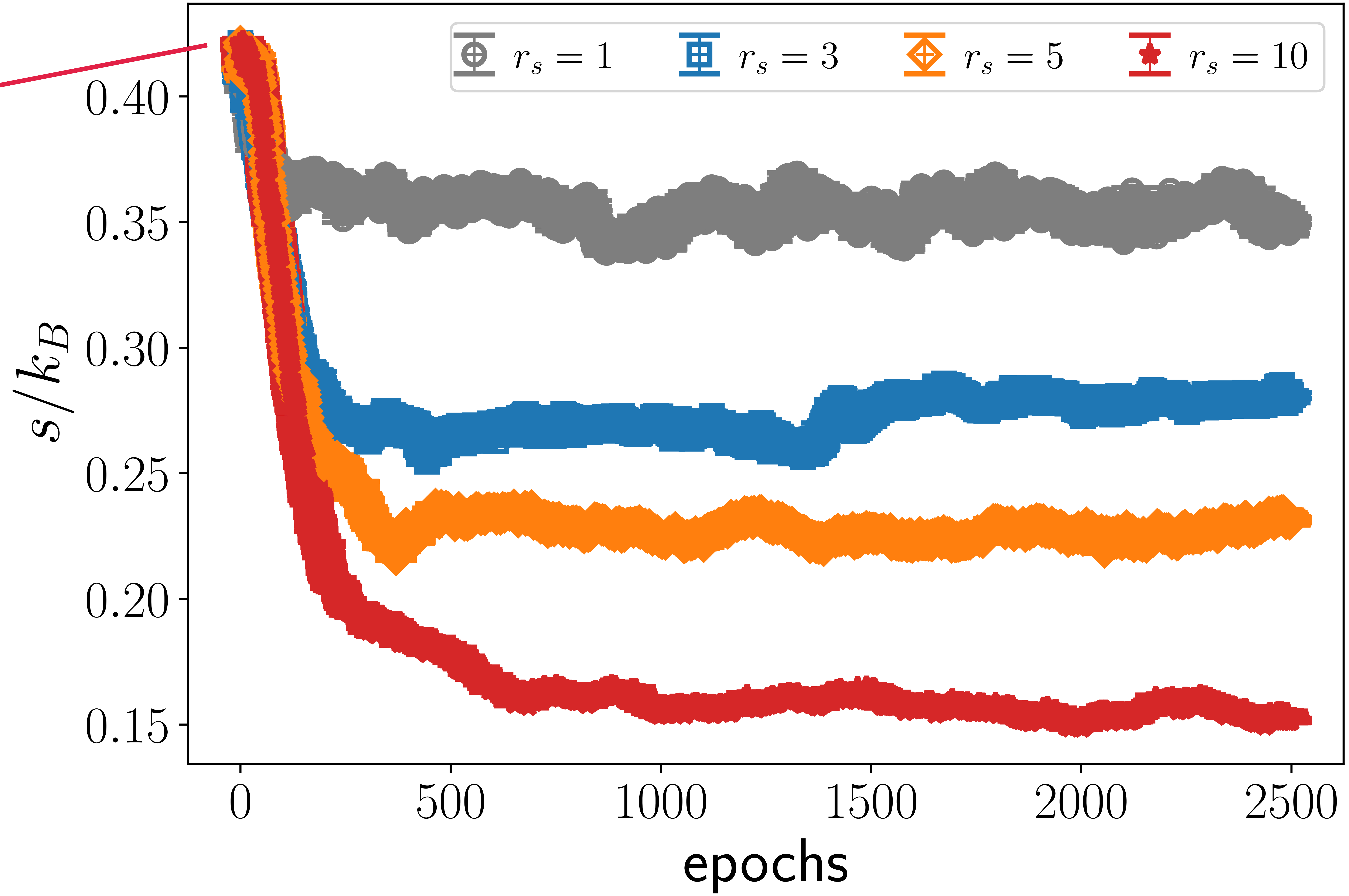
2D electron gas T=0



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

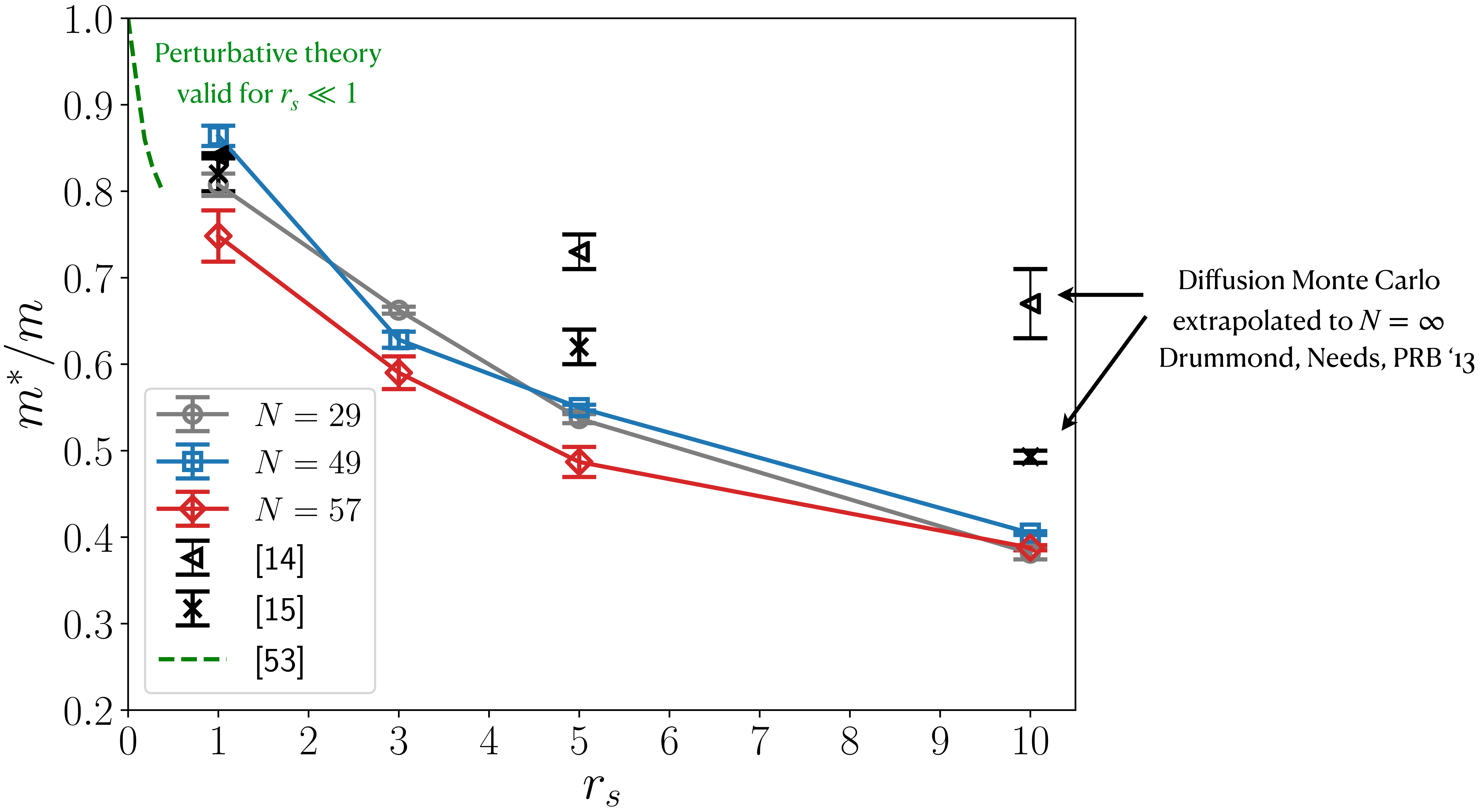
37 spin-polarized electrons in 2D @ T/T_F=0.15

Starting point:
Ideal gas entropy s_0



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

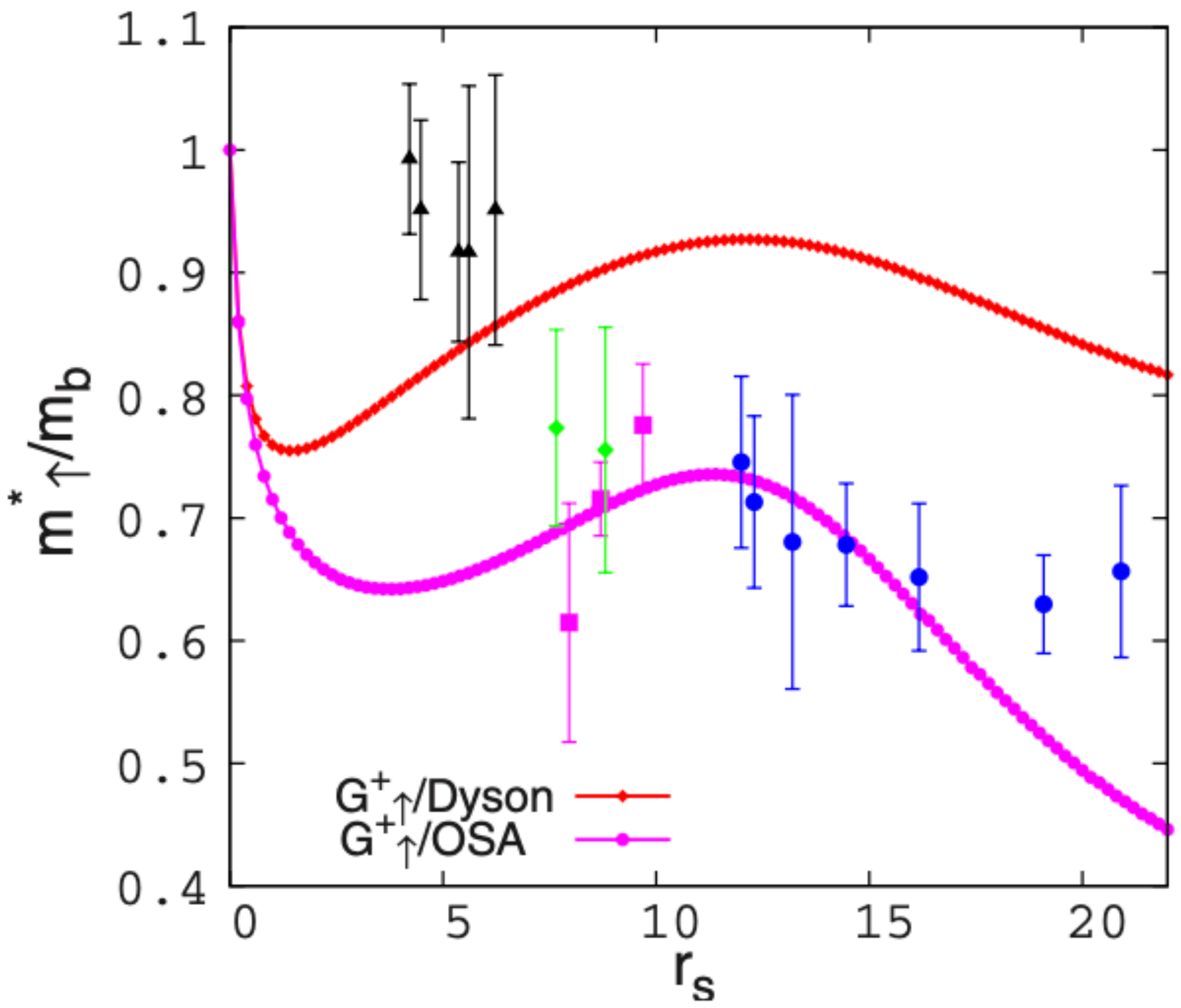
Effective mass of spin-polarized 2DEG



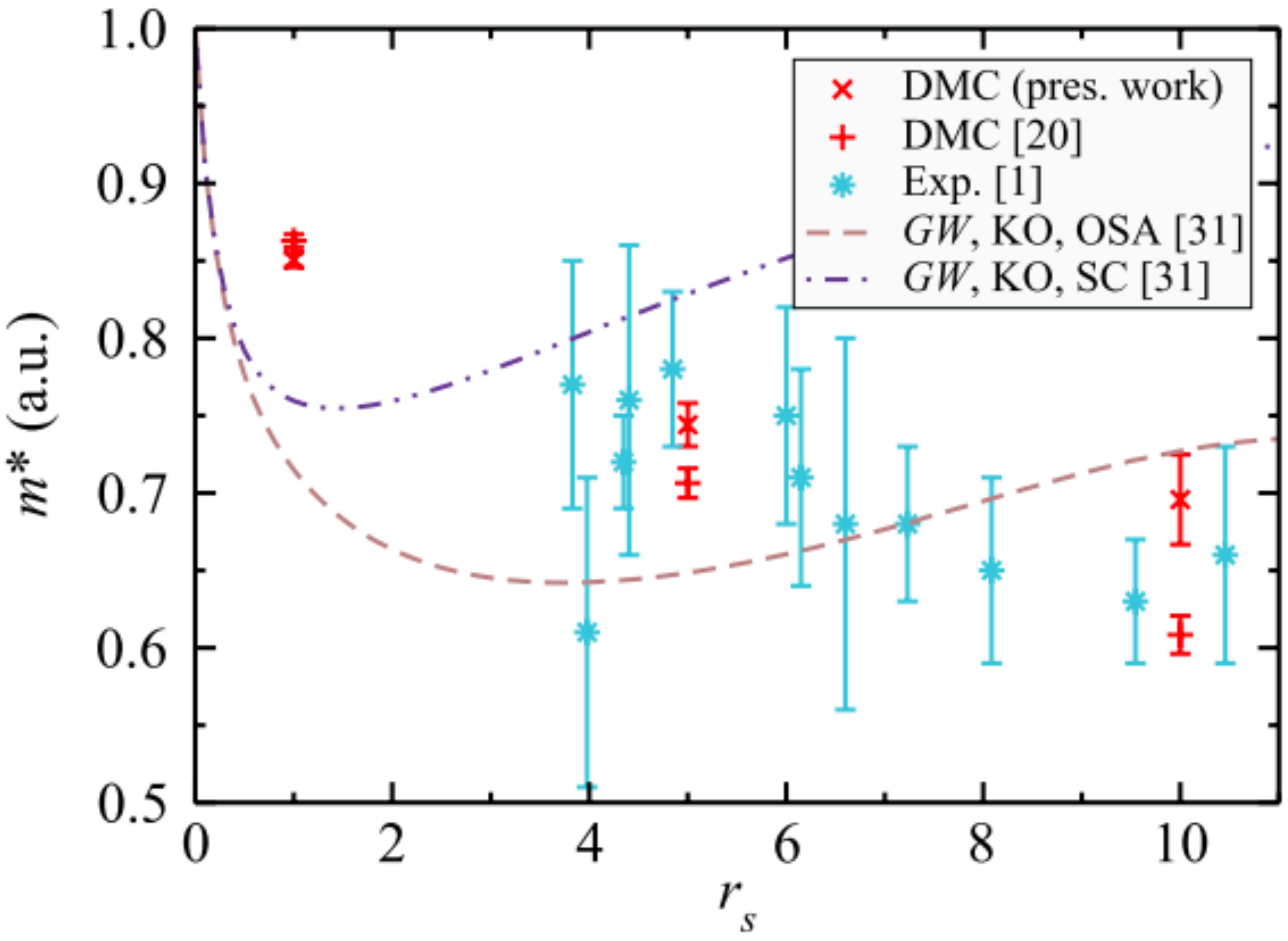
More pronounced suppression of m^* in the low-density strong-coupling region

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

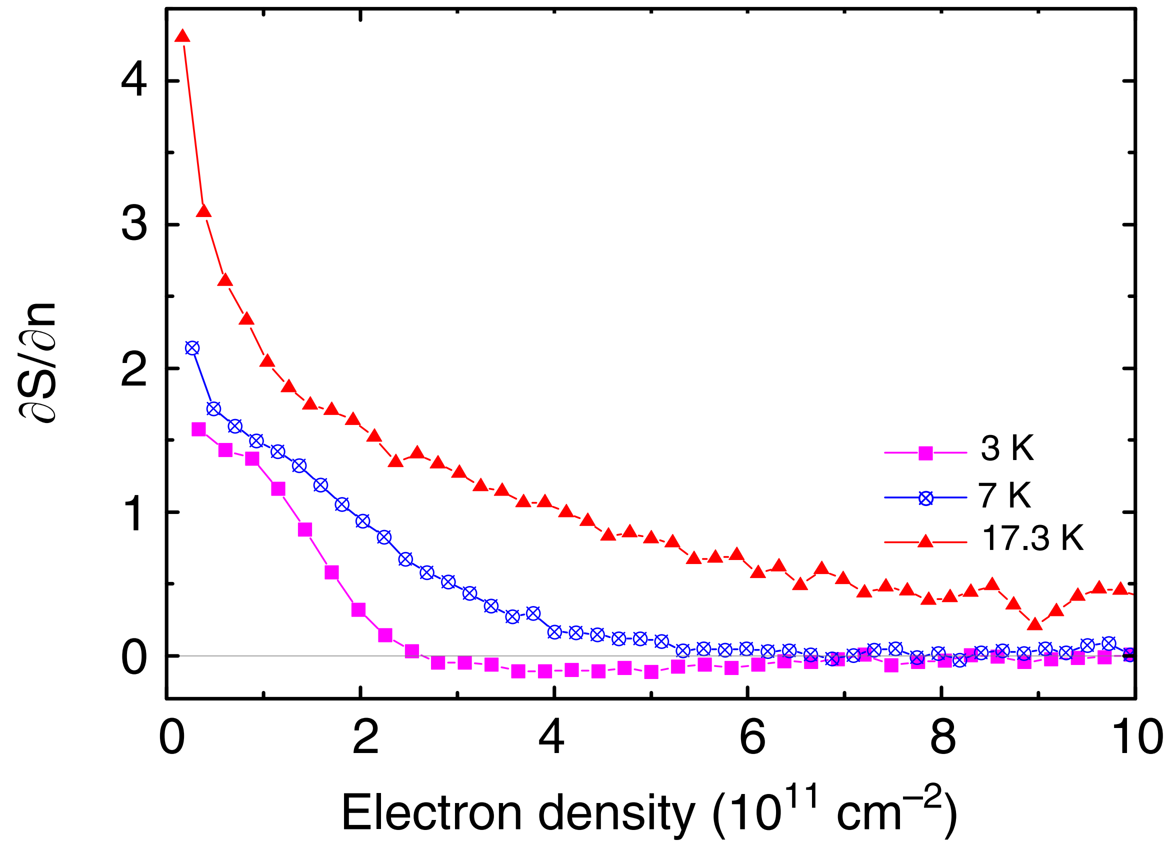
Received 16 May 2014 | Accepted 27 Apr 2015 | Published 23 Jun 2015

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

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

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

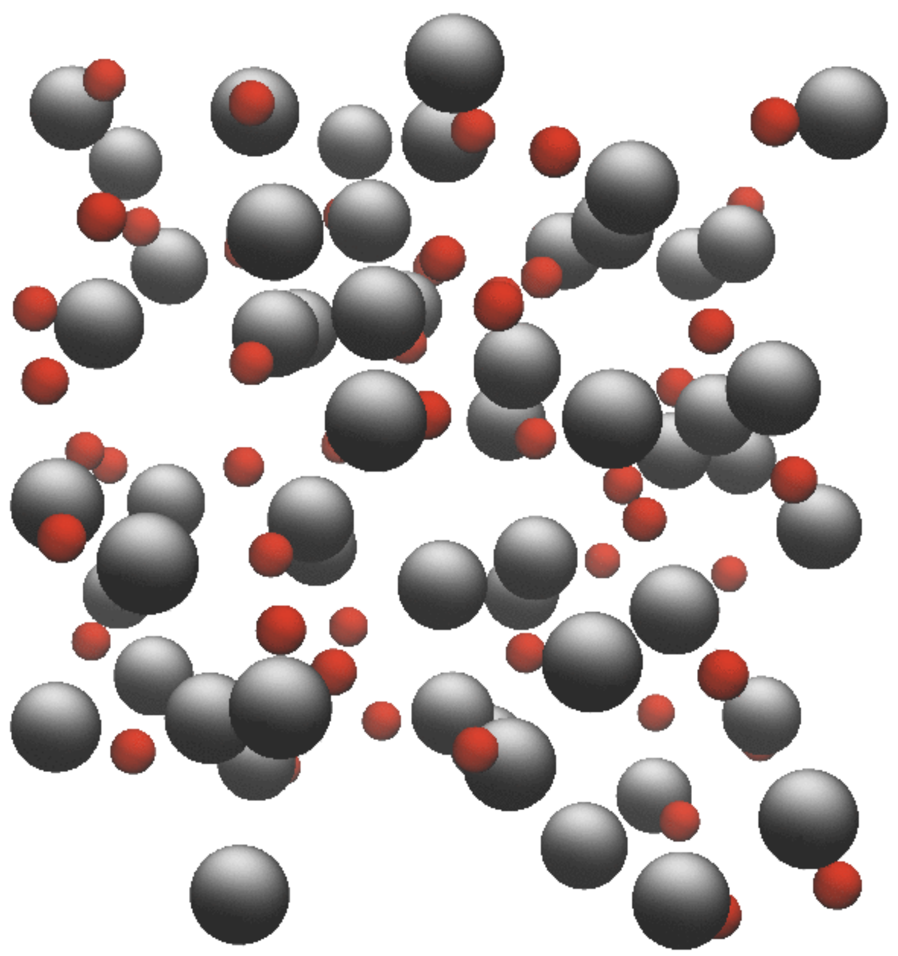
$\zeta \leftrightarrow \boldsymbol{x}$ illustrates adiabatic continuity.

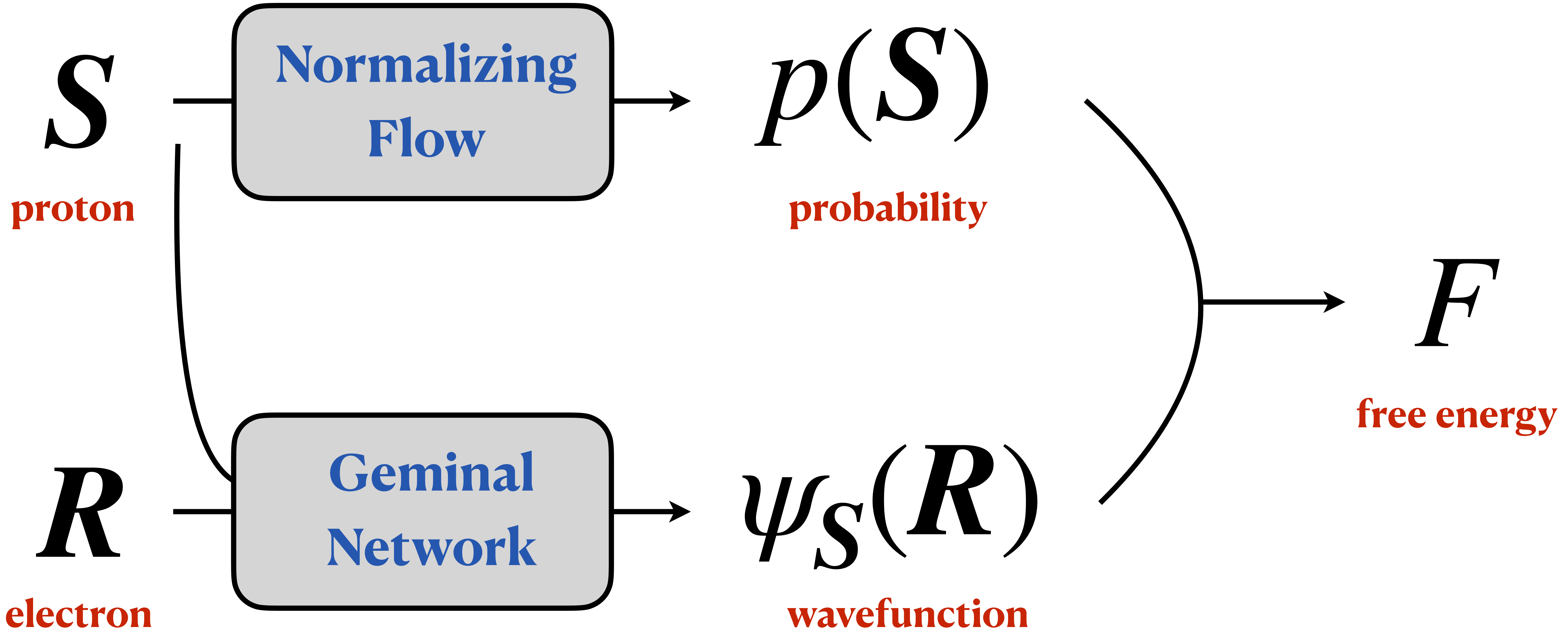
Deep variational free energy for dense hydrogen

Xie, Li, Wang, Zhang, LW, 2209.06095

$$F = \mathop{\mathbb{E}}_{\boldsymbol{S}\sim p(\boldsymbol{S})} \left[k_B T \ln p(\boldsymbol{S}) + \mathop{\mathbb{E}}_{\boldsymbol{R}\sim|\psi_{\boldsymbol{S}}(\boldsymbol{R})|^2} \left[\frac{H\psi_{\boldsymbol{S}}(\boldsymbol{R})}{\psi_{\boldsymbol{S}}(\boldsymbol{R})} \right] \right]$$

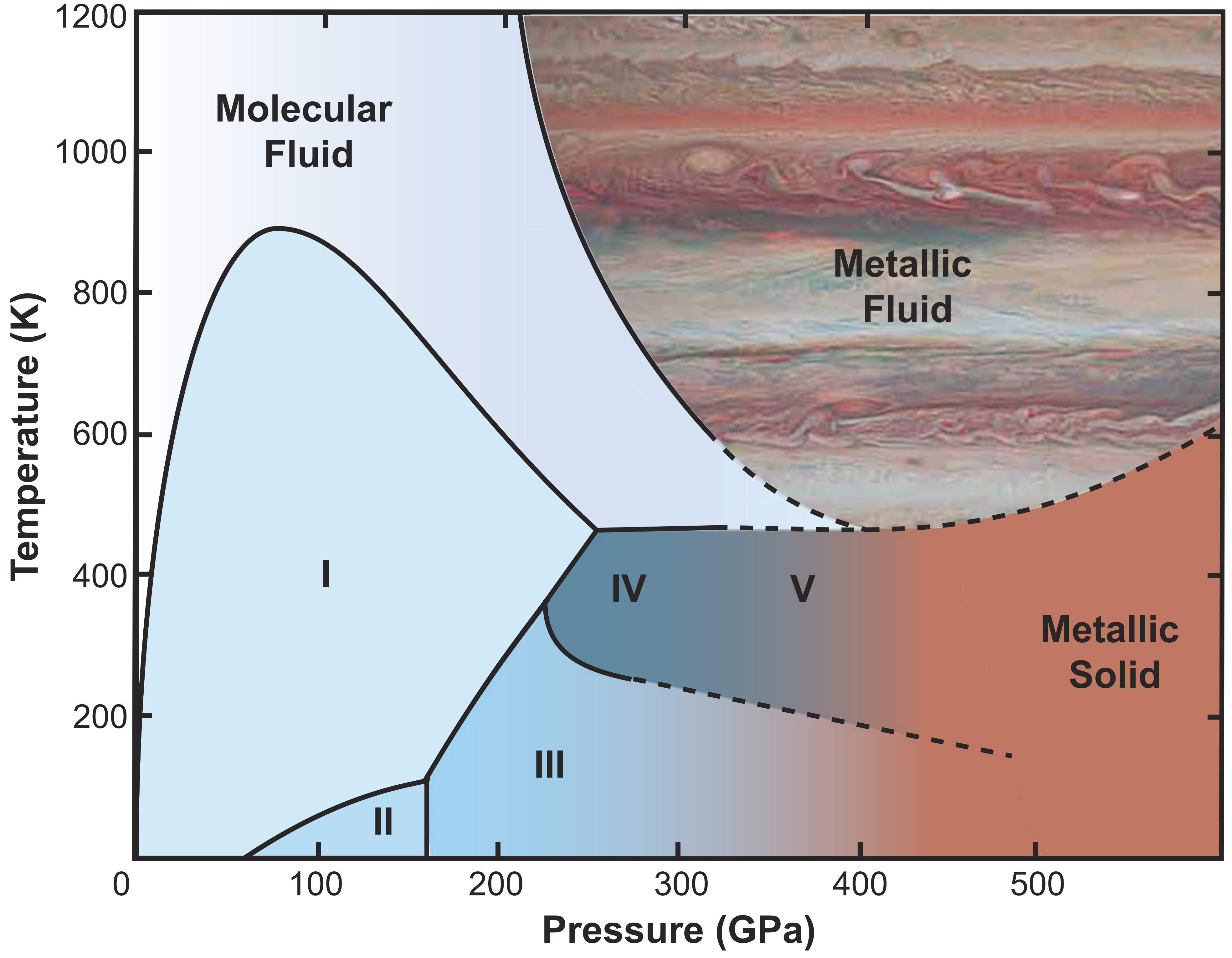
Classical protons coupled to ground state electrons



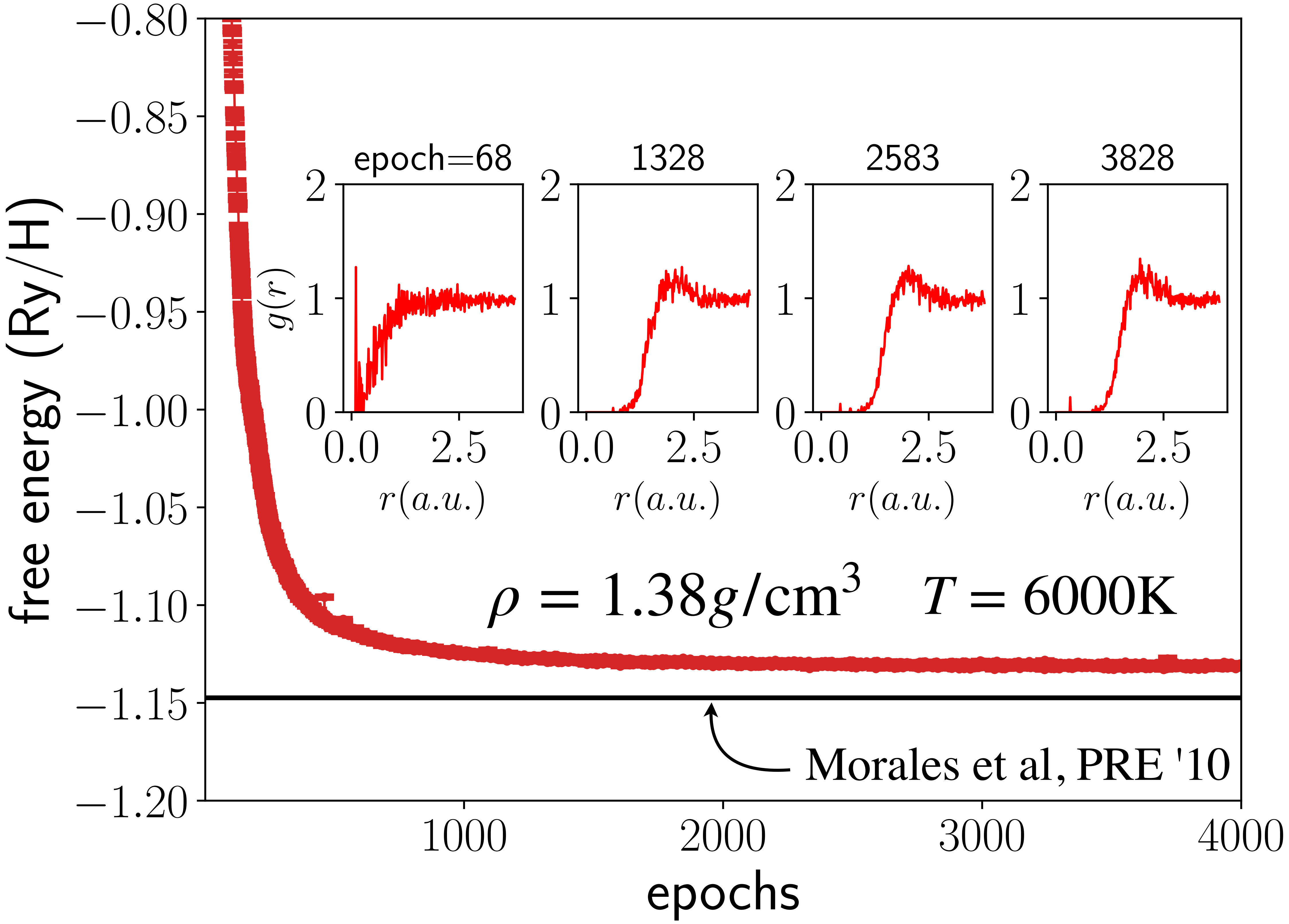


The dense hydrogen problem

Gregoryanz et al, Matter Radiat. Extremes, 2020

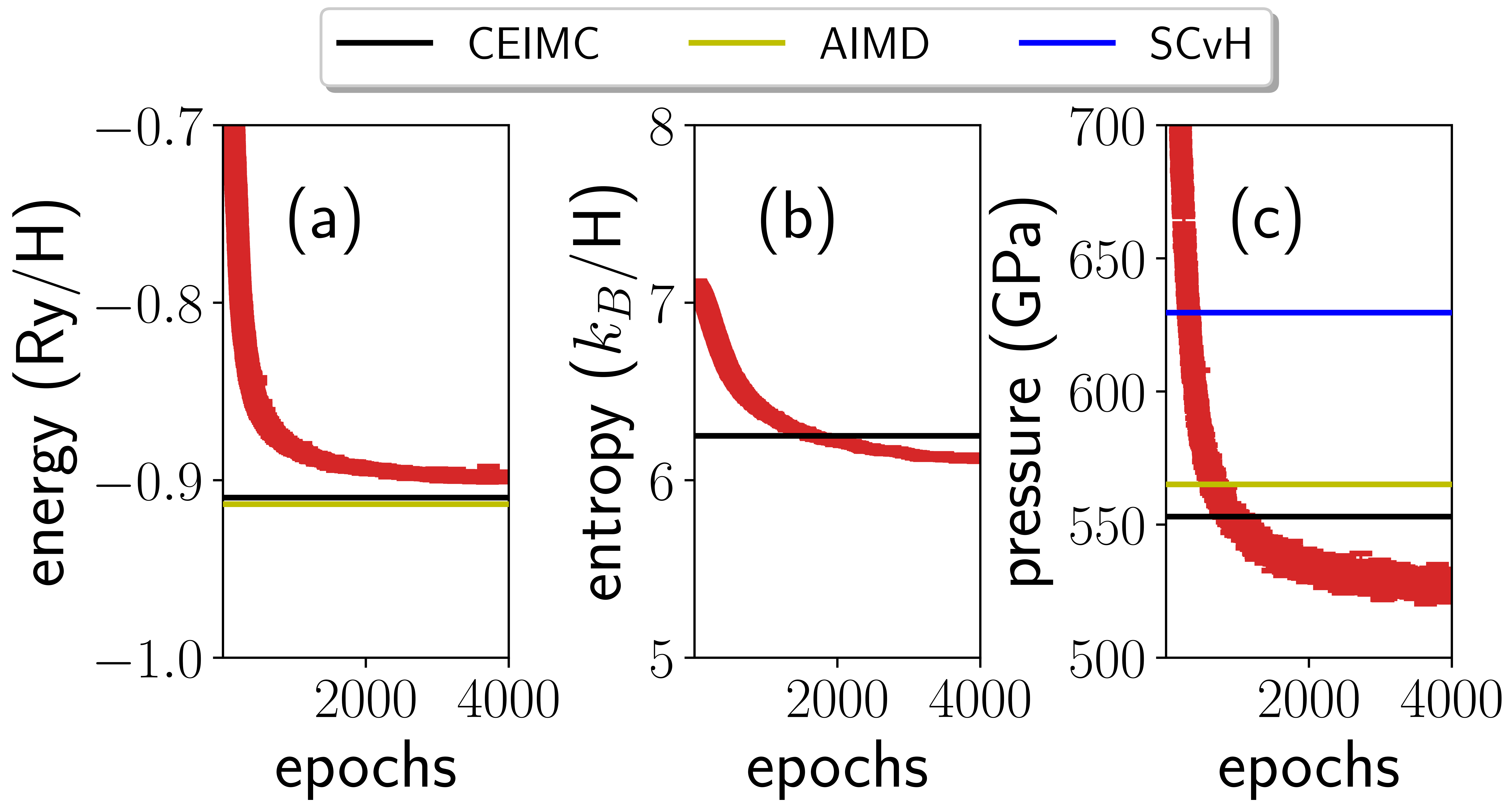


Xie, Li, Wang, Zhang, LW, 2209.06095



Generative model for proton probability density distribution

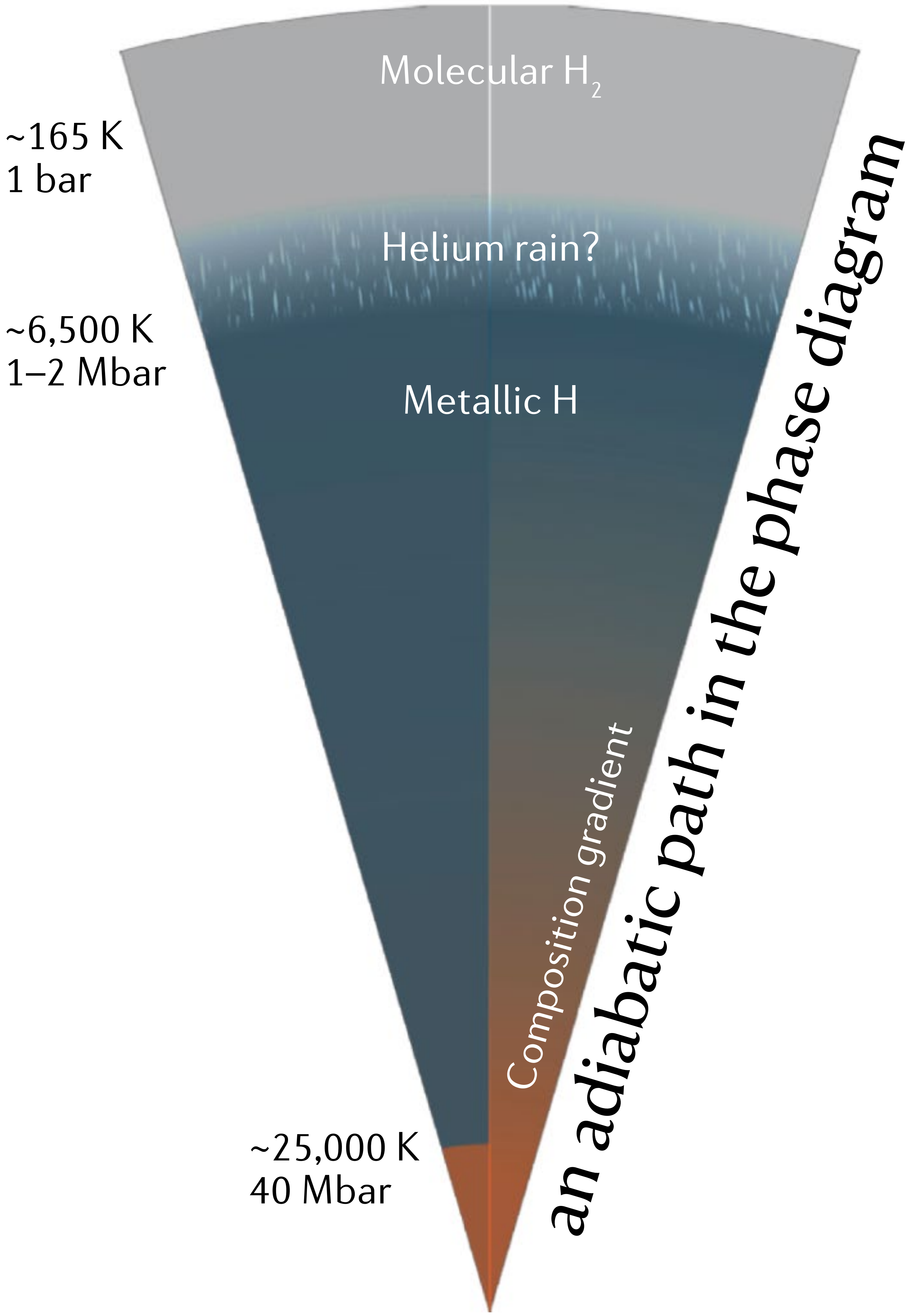
Deep neural network (Ferminet) for electron wavefunction



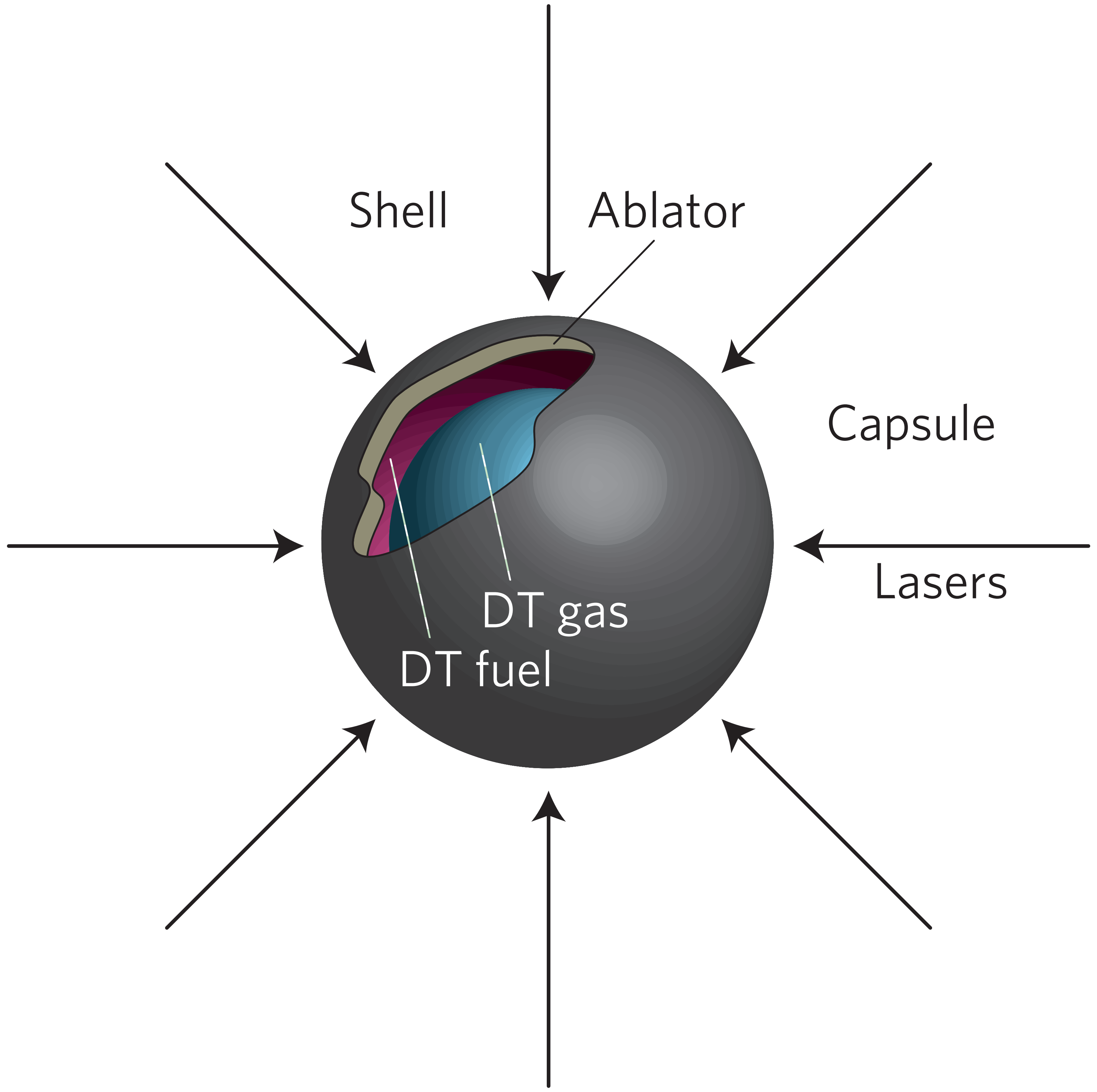
Our predictions will be systematically improved when lowering the variational free energy

Dense hydrogen in the sky and in the lab

Jupiter interior



Inertial confinement fusion



Equation-of-state is the input for hydrodynamics simulations

“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

Thanks to my collaborators

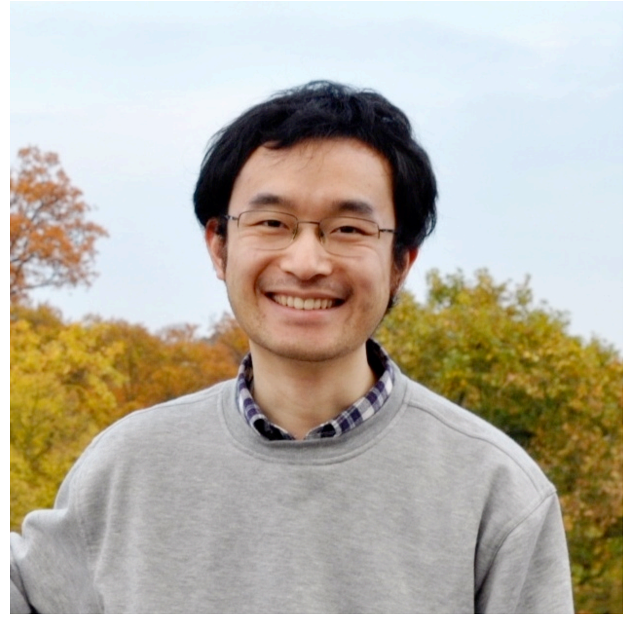
Shuo-Hui Li

Dian Wu

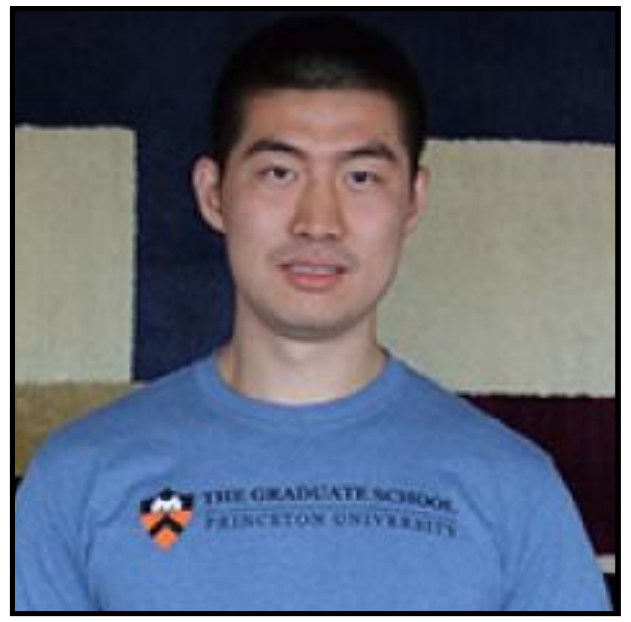
Hao Xie

Zi-Hang Li

Pan Zhang



Han Wang



Linfeng Zhang

1802.02840, PRL '18
1809.10606, PRL '19
2105.08644, JML '22
2201.03156, 2209.06095

li012589/NeuralRG
wdphy16/stat-mech-van
fermiflow/CoulombGas
fermiflow/hydrogen