中国科学院大学 University of Chinese Academy of Sciences
核科学与技术学院 School of Nuclear Science and Technology
RIKEN iTHEMS 理化学研究所 数理創造プログラム RIKEN Interdisciplinary Theoretical and Mathematical Sciences Program
deep-h

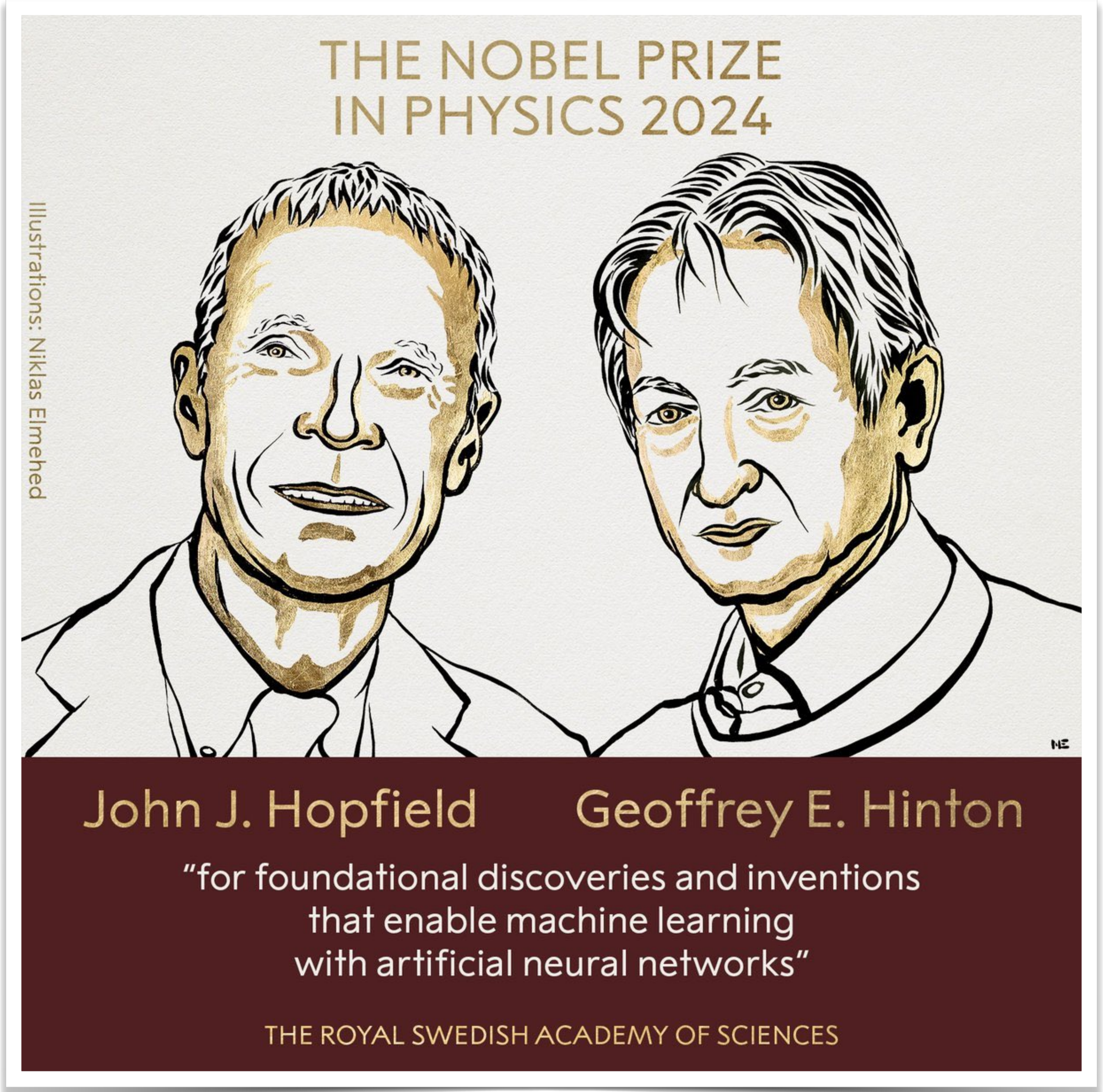
Physics-Driven Learning for Solving Inverse Problems towards QCD Physics

Lingxiao Wang(王凌霄)
RIKEN-**iTHEMS**

核科学与技术学院
School of Nuclear Science and Technology, UCAS
中国科学院大学高能核物理课题组前沿讲座

Arxiv:202412.xxxxx, arxiv:202410.03082, Prog.Part.Nucl.Phys. 104084(2023);
Phys. Rev. D 103, 116023, Phys. Rev. C 106, L051901, Phys. Rev. D 107, 083028, Phys. Rev. D 106, L051502;
Chin. Phys. Lett. 39, 120502, Phys. Rev. D 107, 056001, JHEP05(2024)060.

Nobel Prize in Physics 2024



For physicists,

it is the best of times,

it is the worst of times.

DEEP-IN Working Group

Concept
Activities
Facilitators
Members
Contact

DEEP-IN SCIENCE

深入 科学

CONCEPT

“DEEP learning for INverse problems (DEEP-IN)” in Sciences Working Group

The essence of discovery in sciences has always been rooted in the reverse engineering of natural phenomena and observational data. This paradigm of deducing the underlying laws of nature from observable outcomes forms the cornerstone of our scientific inquiry. **The DEEP-IN working group is established with the recognition that the elucidation of such complex phenomena demands a fusion of physics insights and advanced deep learning methodologies.**

In response to the evolving landscape of scientific research, our objective is to integrate cutting-edge **deep learning techniques, alongside generative models and other advanced statistical learning methods**, into the toolkit of scientists.

The DEEP-IN working group at RIKEN-iTHEMS is dedicated to creating an interdisciplinary platform that harnesses the transformative potential of artificial intelligence(AI). This platform is designed to **tackle inverse problems that span a diverse spectrum of sciences, from biology to physics and more in the future.**

https://sites.google.com/view/deep-in-wg/homepage

iTHEMS 理化学研究所 数理創造プログラム
RIKEN Interdisciplinary Theoretical and Mathematical Sciences Program

About iTHEMS

HOME / About iTHEMS / Working Groups / DEEP-IN Working Group

DEEP-IN Working Group

“DEEP learning for INverse problems (DEEP-IN) in Sciences” working group (April 1st, 2024 -)

Lattice Computations
Gert Aarts, Swansea U.
Takumi Doi, iTHEMS
Andreas Ipp, TU Wien
Tetsuo Hatsuda, iTHEMS
Yan Lyu, iTHEMS

Now mostly physicists **->** **Future** more diverse scientists

BioPhysics: **Catherine Beauchemin,** iTHEMS
Condensed Matter Physics: **Steffen Backes,** iTHEMS
QCD Physics: **Kenji Fukushima,** UTokyo
Nuclear Physics: **Haozhao Liang,** UTokyo
Quantum Computing: **Enrico Rinaldi,** Quantinuum K.K./iTHEMS

Heavy-Ion Collisions
Long-Gang Pang, CCNU
Shuzhe Shi, THU
Kai Zhou, CUHK-ShenZhen

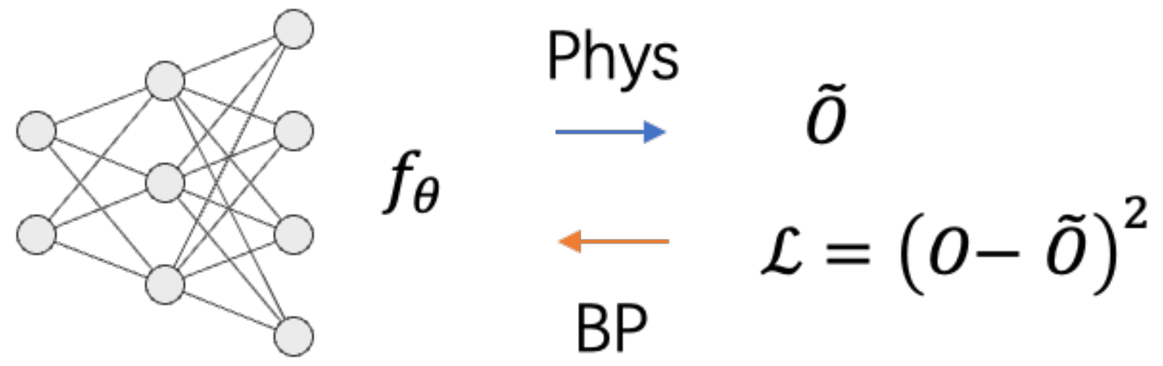
Astrophysics
Márcio Ferreira, Coimbra U.
Yuki Fujimoto, INT->iTHEMS
Akira Harada, NIT-Ibaraki
Zhenyu Zhu, TDLI->RIT

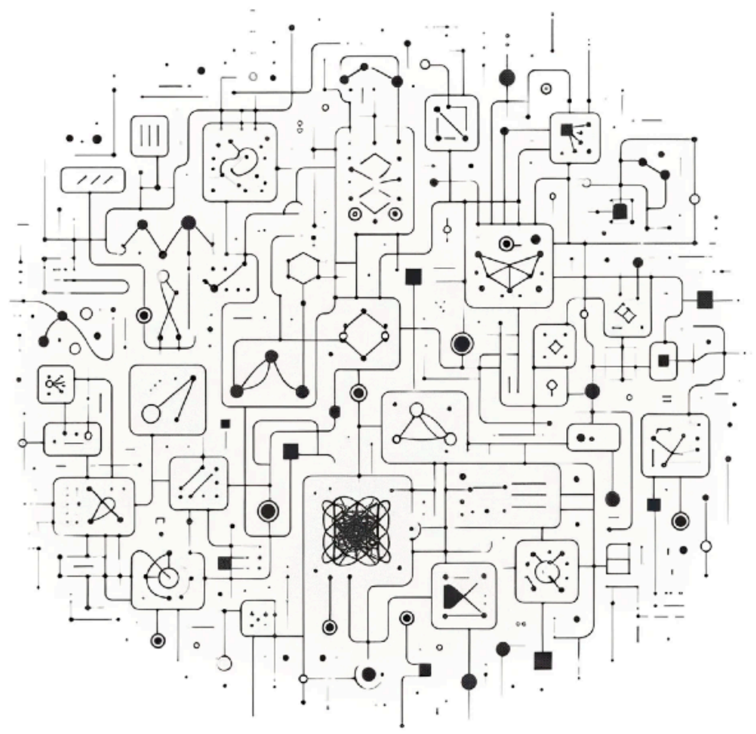
Machine Learning
Akinori Tanaka, AIP/iTHEMS
Lingxiao Wang, iTHEMS

Lingxiao Wang (RIKEN iTHEMS) *Contact at lingxiao.wang@riken.jp

Outline

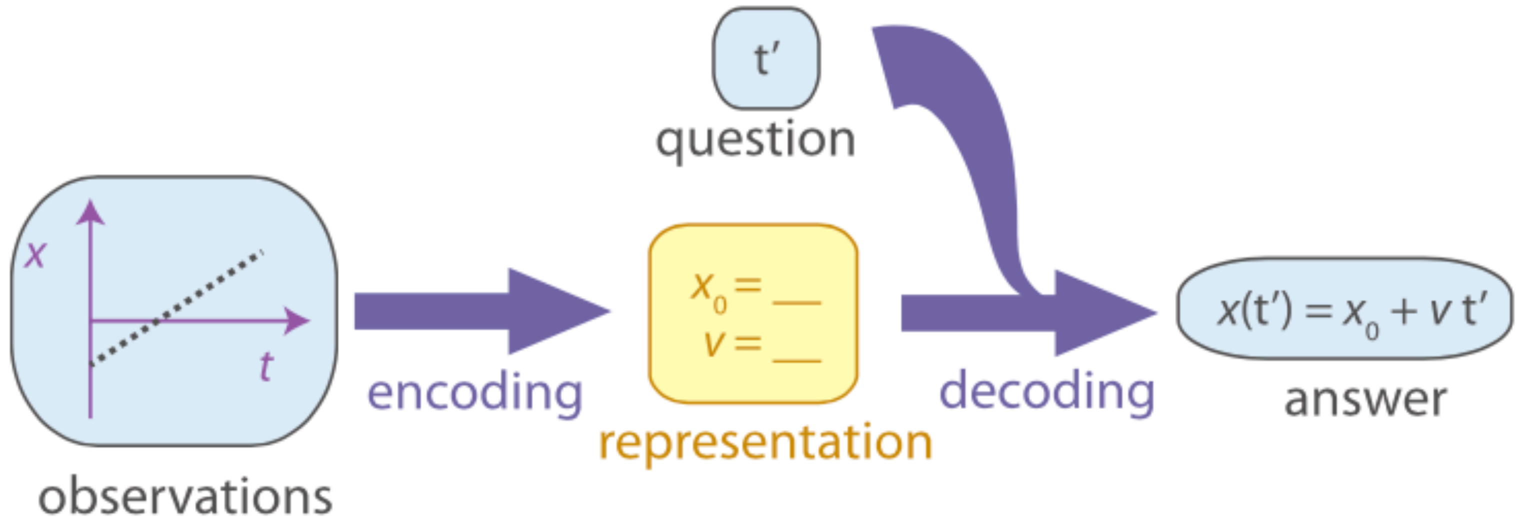
- **Physics for Machine Learning**
- **Inverse Problems**
  - Data-Driven Learning
  - Physics-Driven Learning
- **Generative Models**
  - Learn to Sample
  - Physics-Driven Generative Models
- **Outlooks**



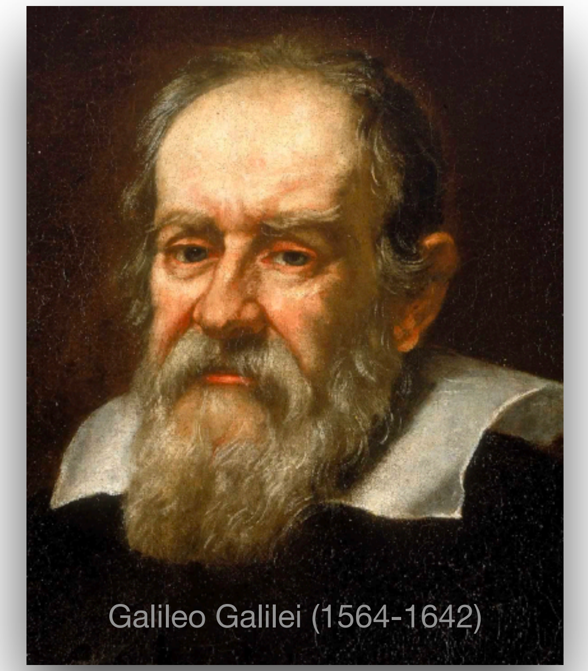


Generated by ChatGPT-4 + DALL·E

Machine Learning and Physics

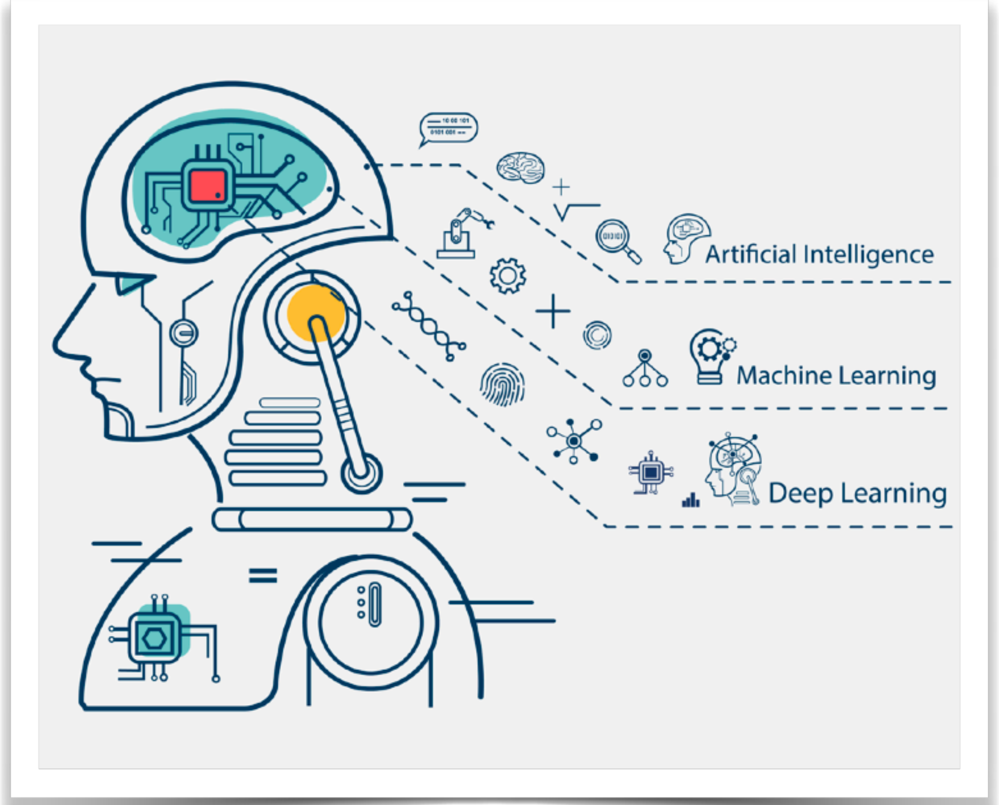


Phys.Rev. Lett. **124**, 010508 (2020)



An inverse problem in science is the process of inferring from a set of observations the causal factors that produced them.

Data, X



Machine, $\{\theta\}$

Prediction $p(X|\theta)$

Estimation

$$\hat{\theta} = \textbf{arg max}_\theta\{p(X|\theta)\}$$

Machine Learning and Inference

Maximum Likelihood Estimation(MLE)

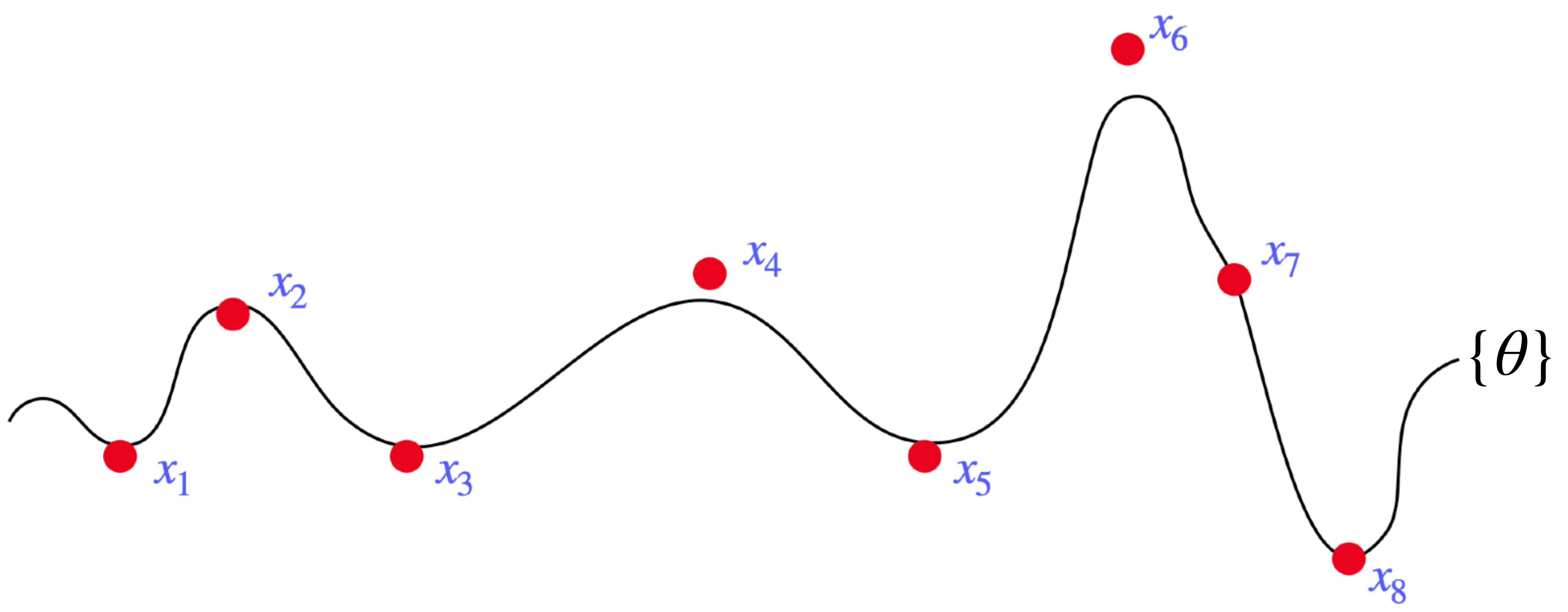
$$\max_{\theta} \prod_{i=1}^{N} p(\mathbf{x}_i \mid \theta)$$

Bayesian Inference

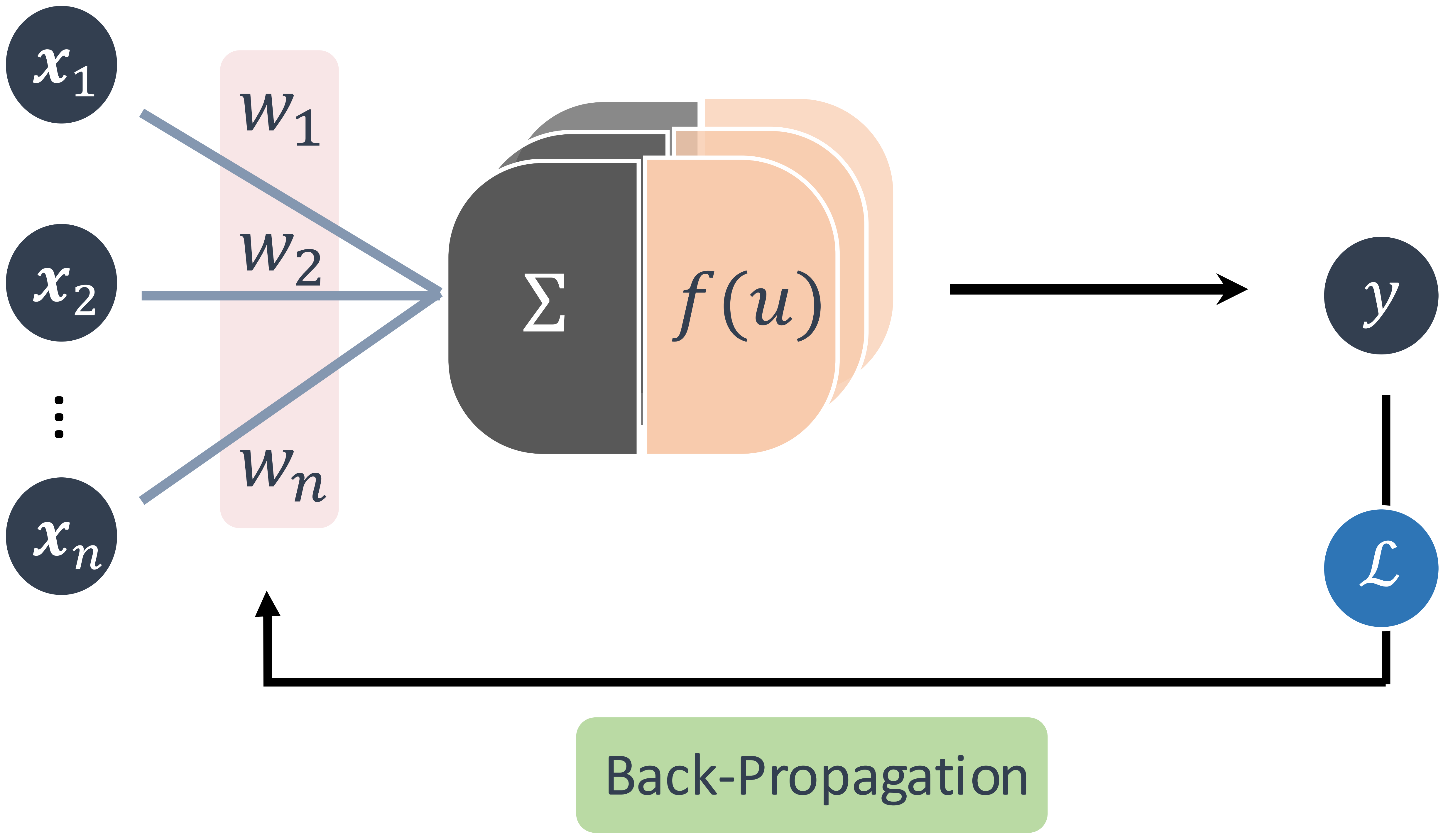
Maximum A Posterior(MAP)

$$p(\theta \mid X) = \frac{p(X \mid \theta)\pi(\theta)}{p(X)}$$

Posterior $p(\theta \mid X)$, **Prior** $\pi(\theta)$, **Evidence** $p(X)$



Deep Model as Machine



Deep (neural network) Model

- **Inputs,** $\{x\} = x_1, x_2, \ldots, x_n$
- **Weights,** $\{w\} = w_1, w_2, \ldots, w_n$
- **Outputs,** y
- **Summation,** $\Sigma(\cdot)$
- **Non-Linear Activation Functions,** $f(u)$
- **Single Layer** $y = f(\Sigma_{i=1}^{n} x_i w_i)$

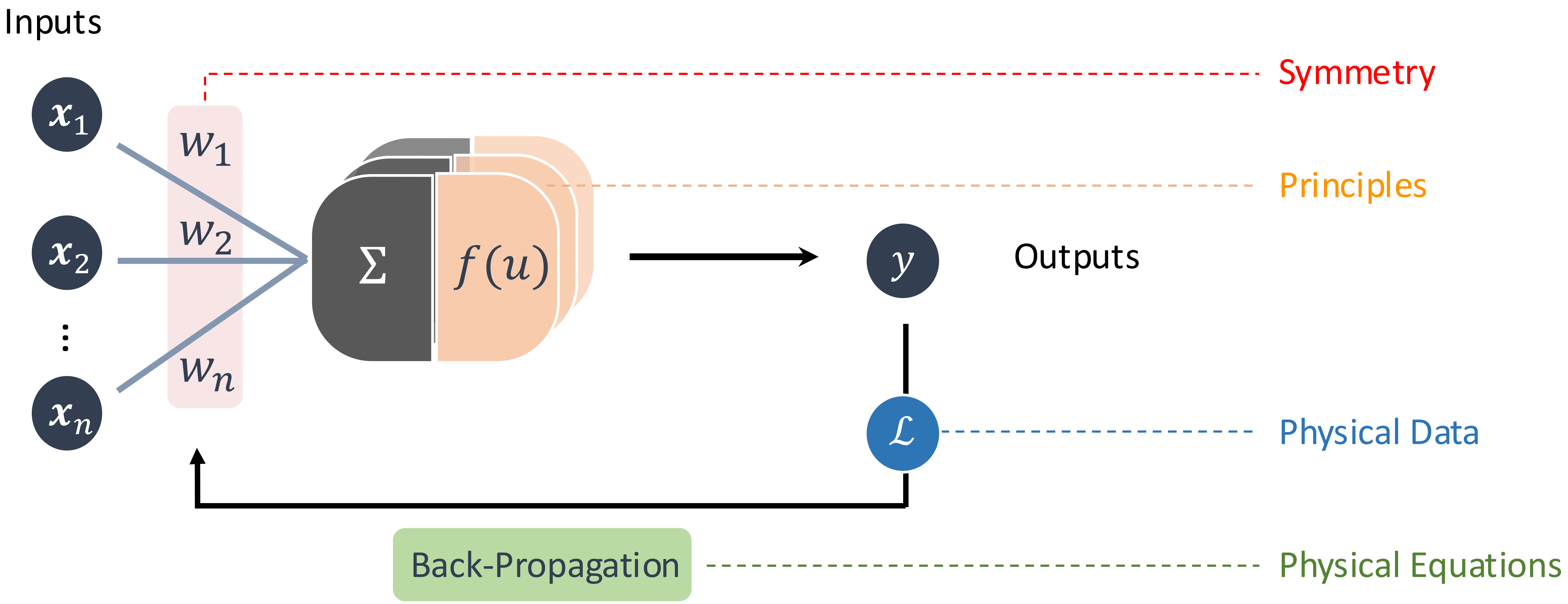
Objective

- **Loss Function,** $\mathscr{L}(y, \hat{y})$
- **Data,** $\hat{y}$

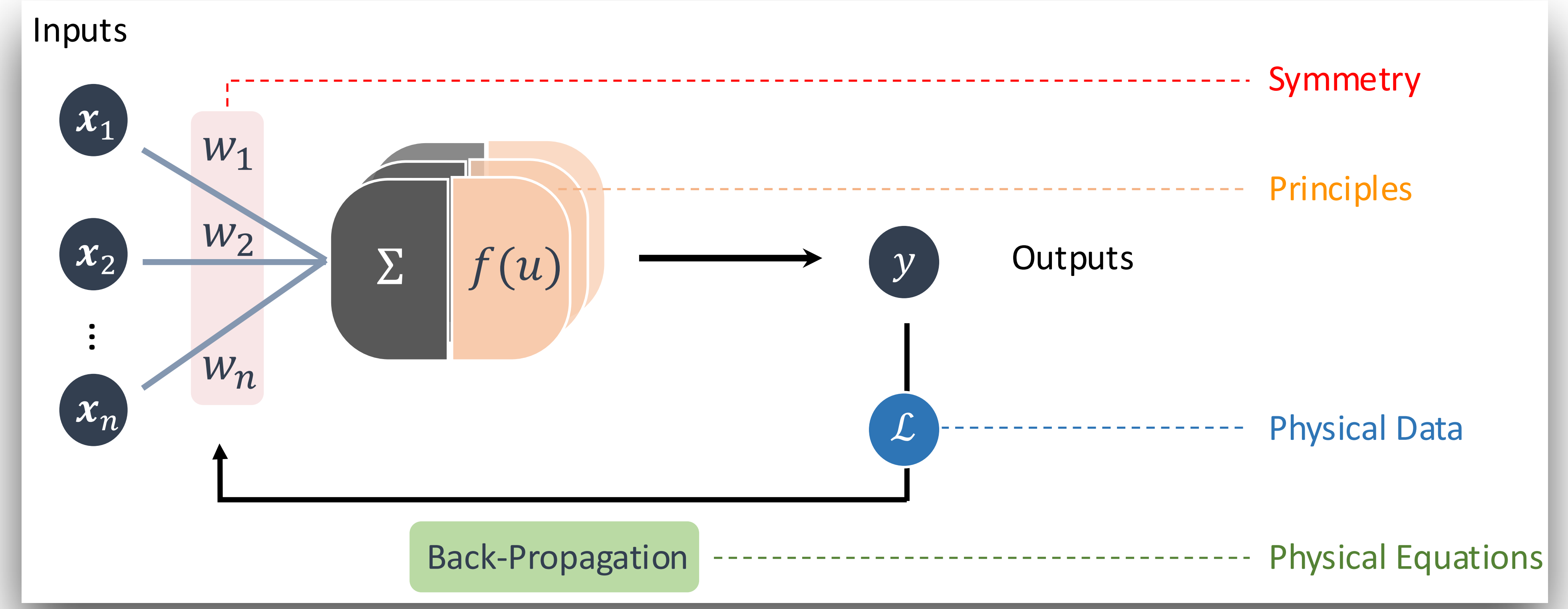
Optimization Algorithm

- **Back-Propagation,** $\frac{\partial \mathscr{L}}{\partial \omega}$
- **Stochastical Methods: SGD, Adam, …**

Physics-Driven Desgins

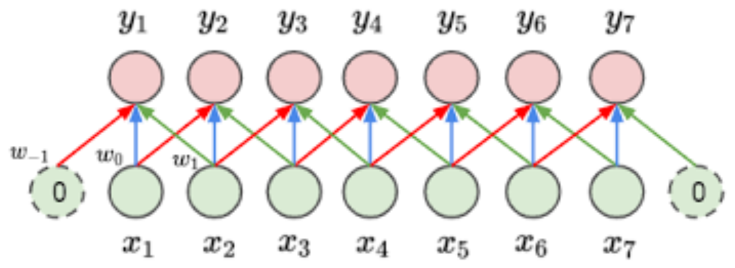


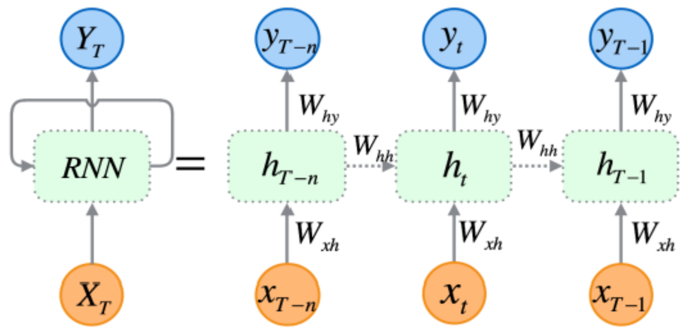
Physics-Driven Desgins: Symmetry

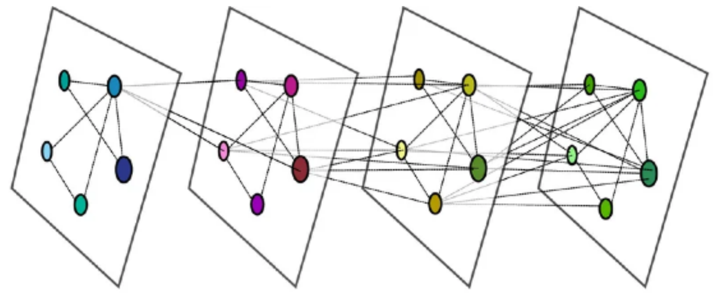


Embedding symmetries introduces a scheme for sharing parameters in deep models

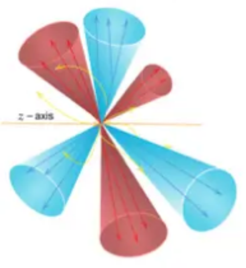
Convolutional Neural Networks (CNNs)	Recurrent Neural Networks (RNNs)	Graph Neural Networks (GNNs)	Equivariant Neural Networks (G-CNNs)
$Y_{i,j,c} = \sum_{m=1}^{k_h}\sum_{n=1}^{k_w}\sum_{p=1}^{C_{in}} X_{i+m,j+n,p}\cdot K_{m,n,p,c}+b_c$	$h_t = \sigma(W_x x_t + W_h h_{t-1} + b)$	$h_i^{(l+1)} = \sigma\left(W^{(l)}\cdot\sum_{j\in\mathcal{N}(i)} h_j^{(l)} + b^{(l)}\right)$	$Y_g = \sum_{m=1}^{k_h}\sum_{n=1}^{k_w} X_g(i+m,j+n)\cdot K(m,n)$
translational symmetry	**temporal symmetry**	**permutational symmetry**	**groups**



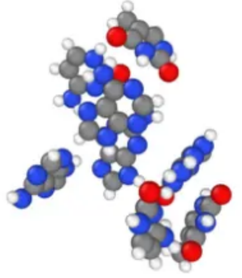




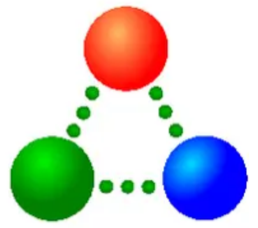
$SO(1,3)$



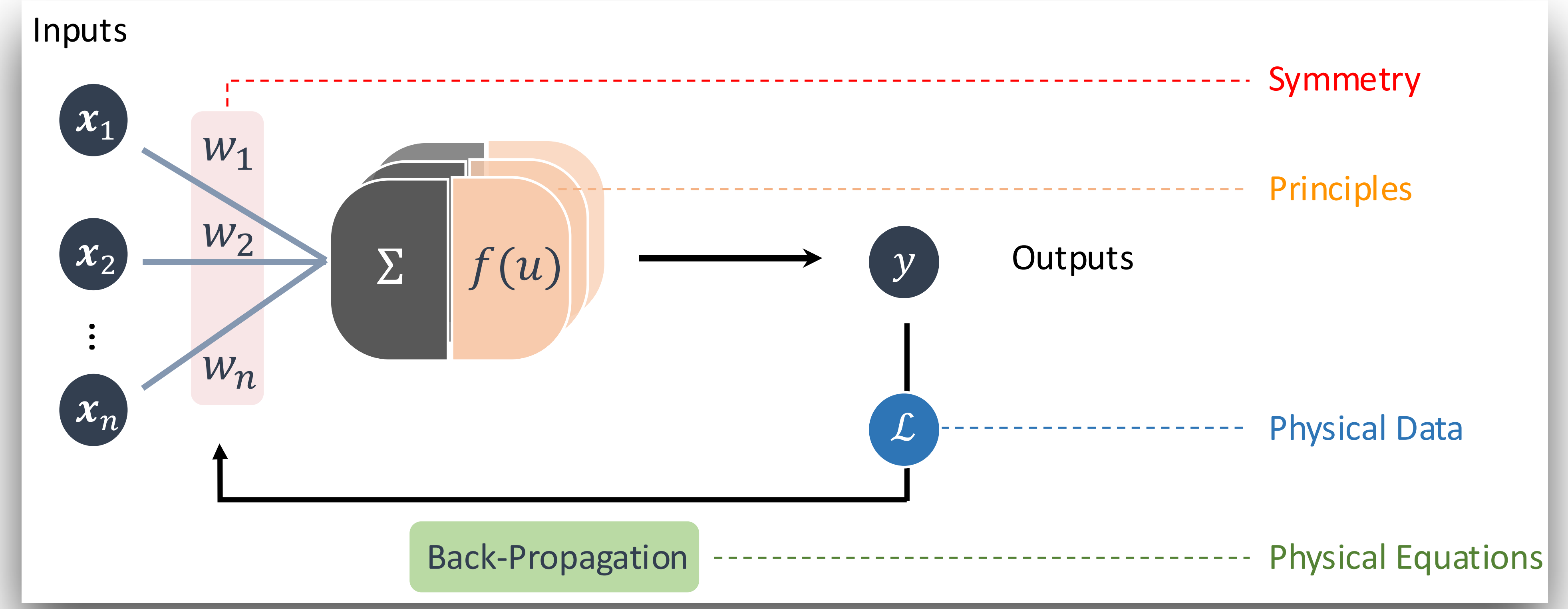
$E(3)$



$SU(3)$



Physics-Driven Desgins: Symmetry



Embedding symmetries introduces a scheme for sharing parameters in deep models

Convolutional Neural Networks (CNNs)

$$Y_{i,j,c} = \sum_{m=1}^{k_h}\sum_{n=1}^{k_w}\sum_{p=1}^{C_{in}} X_{i+m,j+n,p}\cdot K_{m,n,p,c} + b_c$$

- X : Input feature map.
- K : Convolution filter.
- Y : Output feature map.

Recurrent Neural Networks (RNNs)

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b)$$

- x_t : Input at time t.
- h_t : Hidden state at time t.
- W_x and W_h : Shared weight matrices across time.

Graph Neural Networks (GNNs)

$$h_i^{(l+1)} = \sigma\left(W^{(l)}\cdot \sum_{j\in\mathcal{N}(i)} h_j^{(l)} + b^{(l)}\right)$$

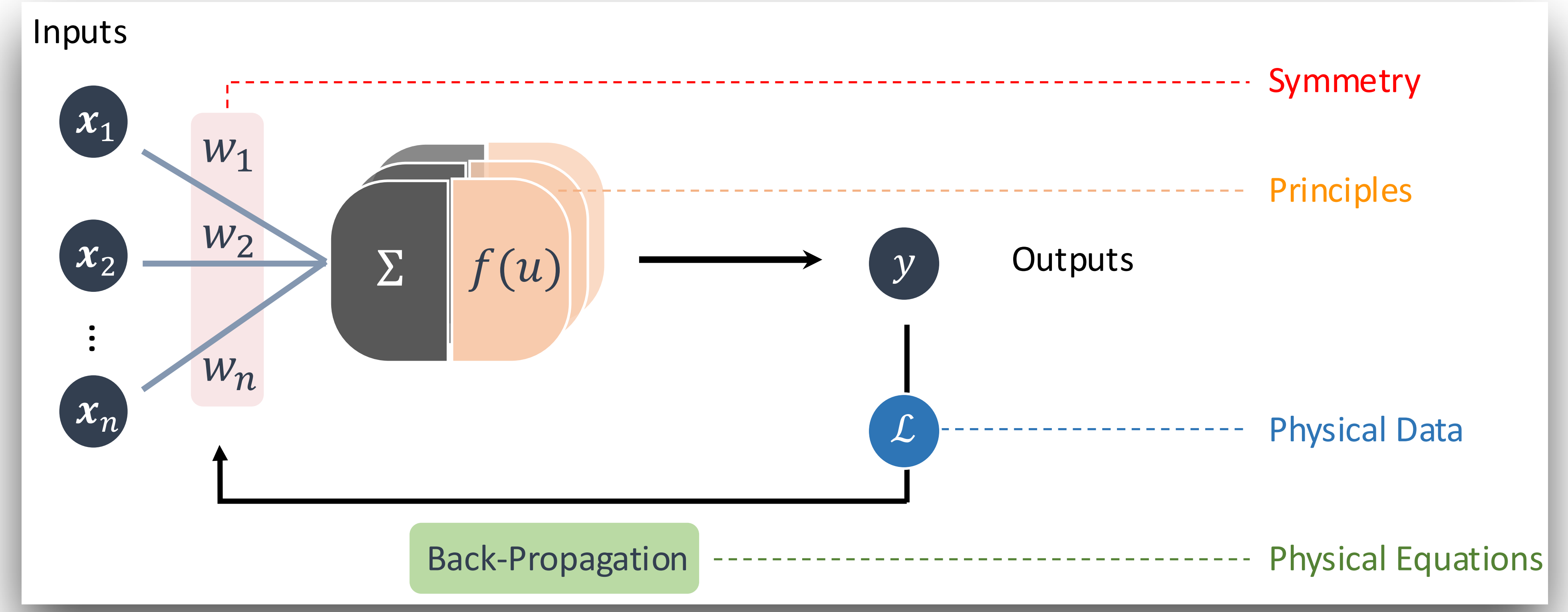
- $h_i^{(l)}$: Feature vector for node i at layer l.
- $W^{(l)}$: Shared weight matrix across all nodes.

Equivariant Neural Networks (G-CNNs)

$$Y_g = \sum_{m=1}^{k_h}\sum_{n=1}^{k_w} X_g(i+m, j+n)\cdot K(m,n)$$

- $g \in G$: Transformation from the group G (e.g., rotations).
- X_g : Input after applying transformation g.

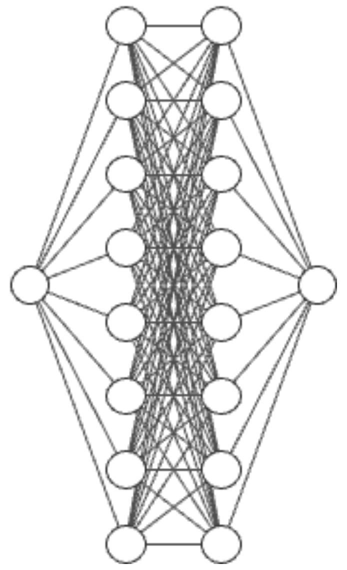
Physics-Driven Desgins: Principles



Principles can enforce the outputs are physically meaningful

Continuity

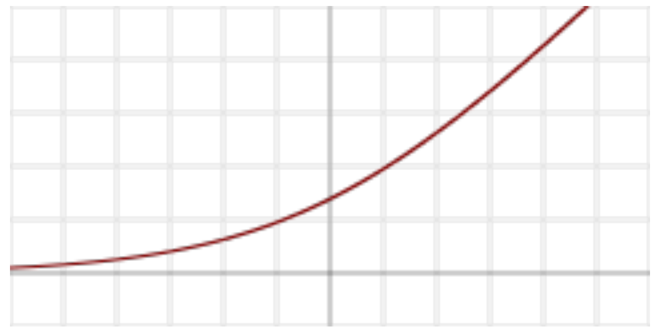
$$y := f(x)$$



Positive Definiteness

Softplus

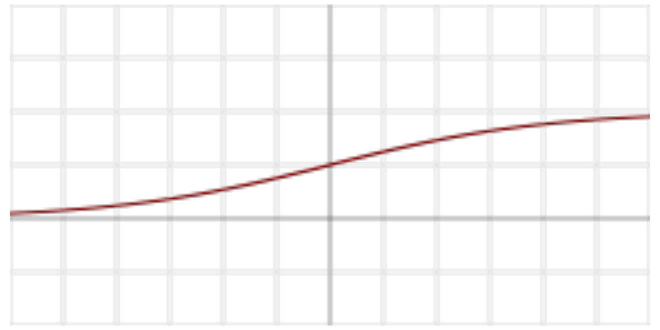
$$\ln(1 + e^x)$$



Causality

Sigmoid $\dfrac{1}{1 + e^{-x}}$

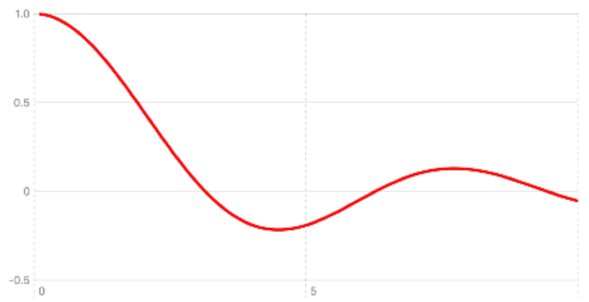
$$\frac{dp}{d\epsilon} = \frac{c_s^2}{c^2} < 1$$



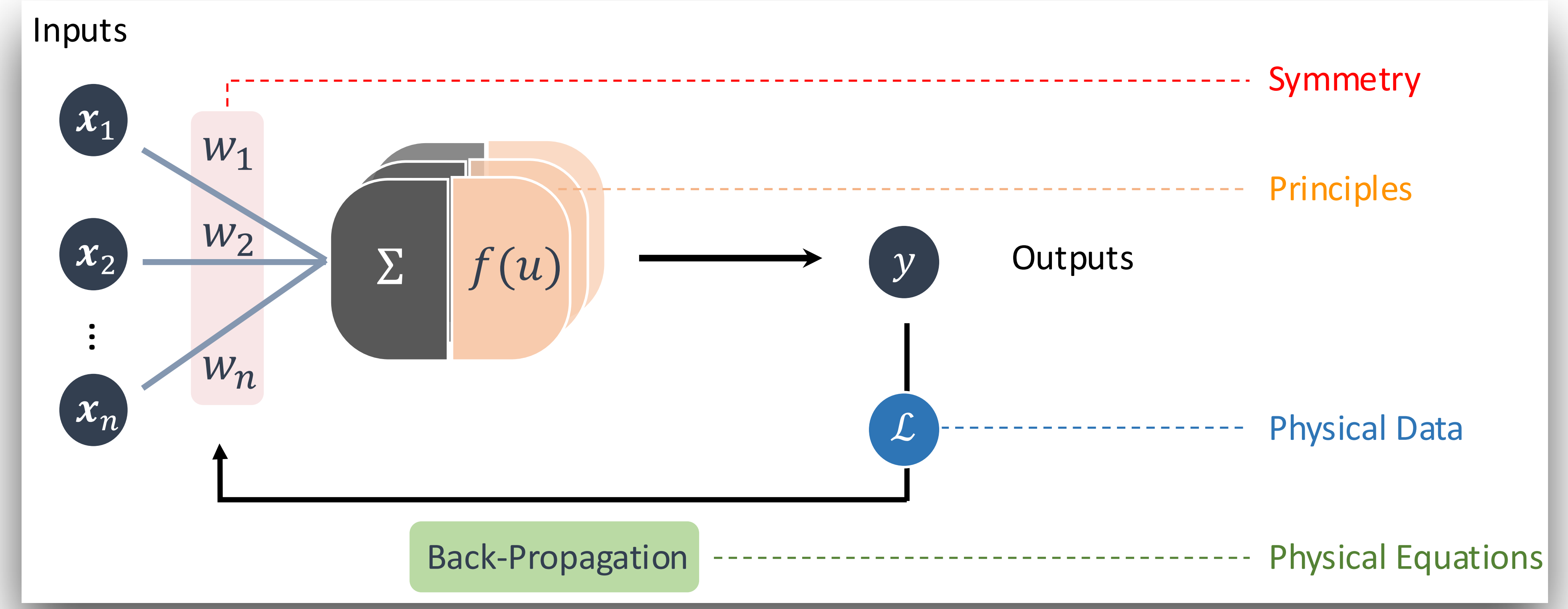
Asymptotic Behaviors

Scattering Wave Function

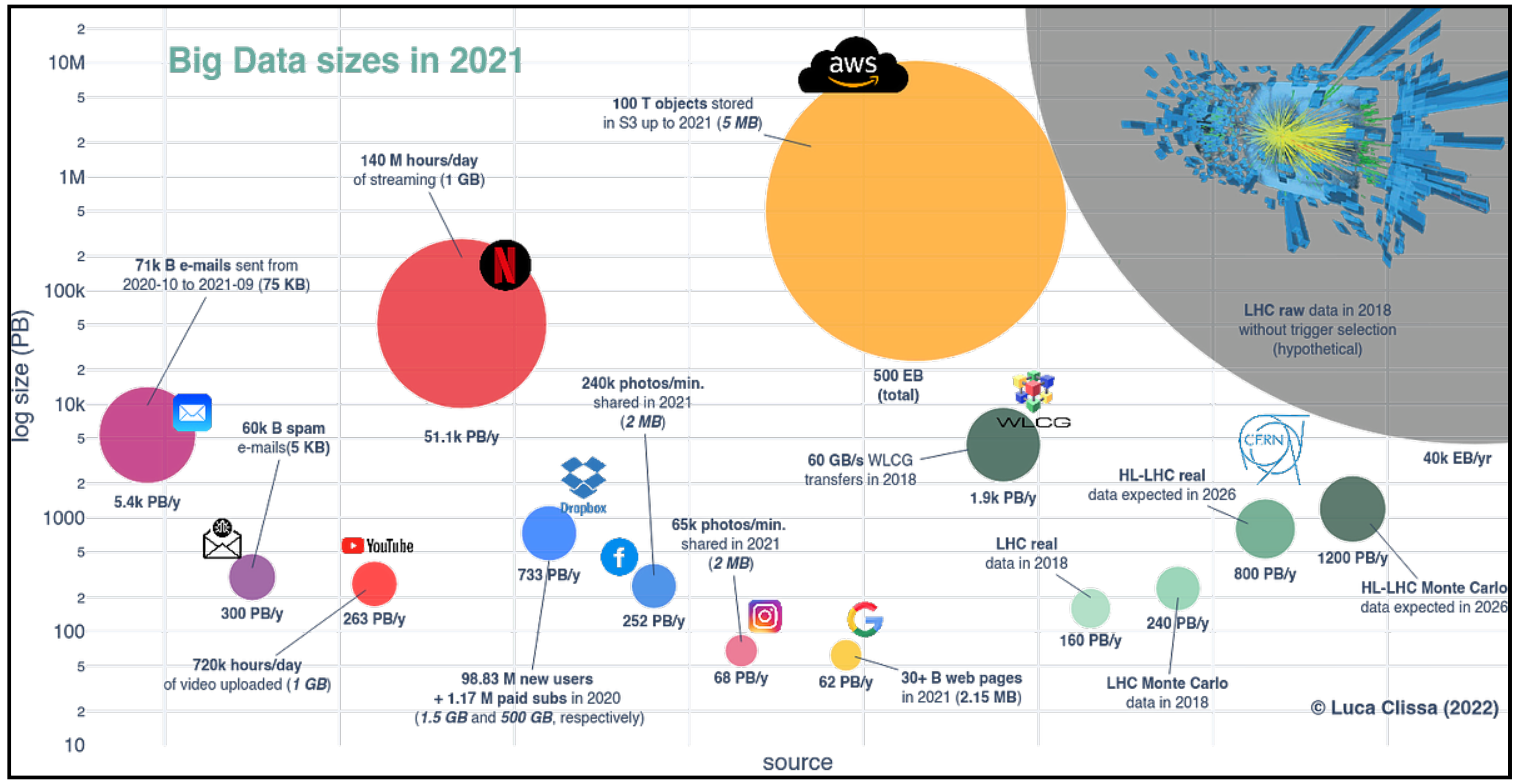
$$A\frac{\sin(kr)}{r} + f(r)$$

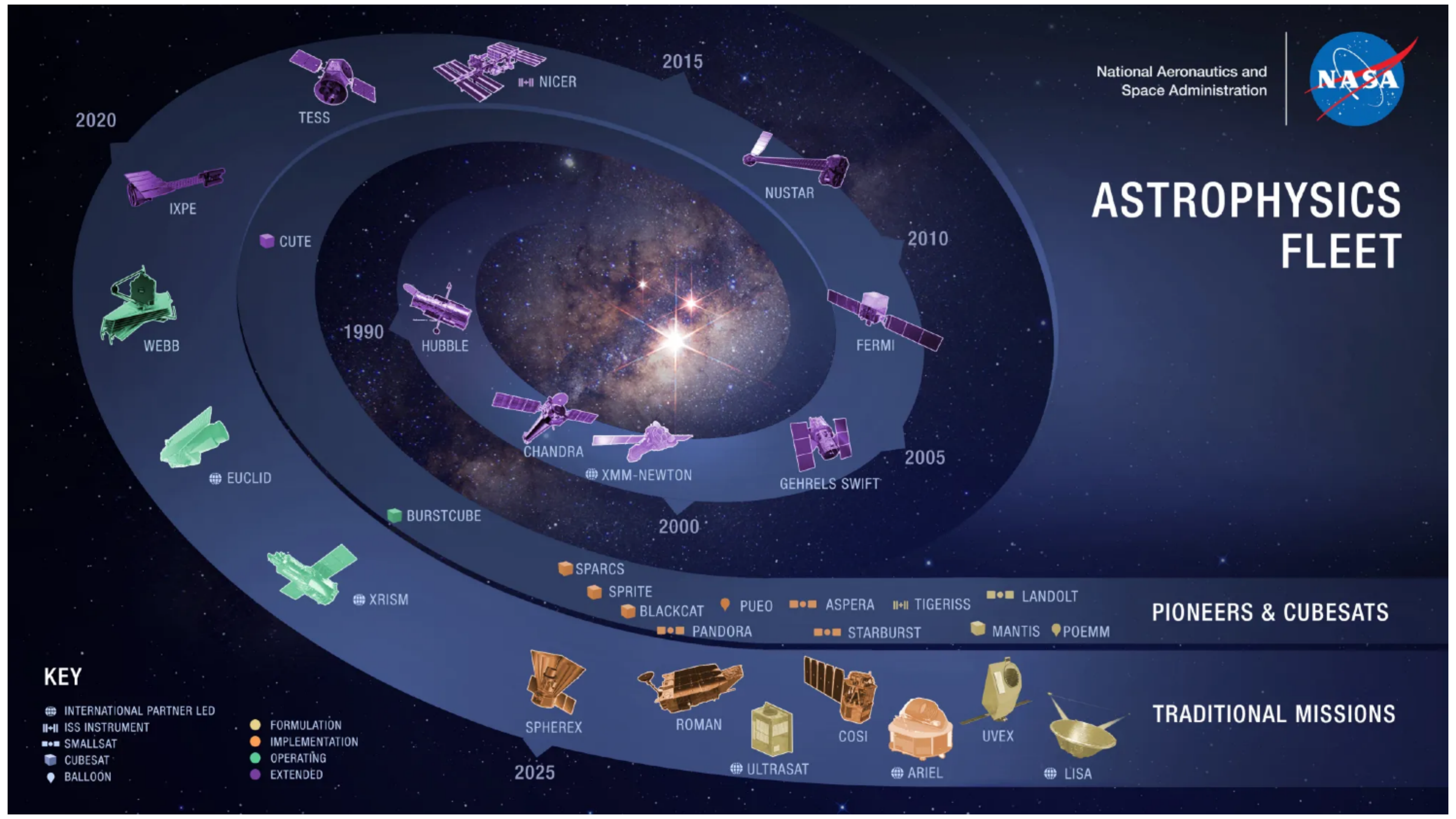


Physics-Driven Desgins: Data

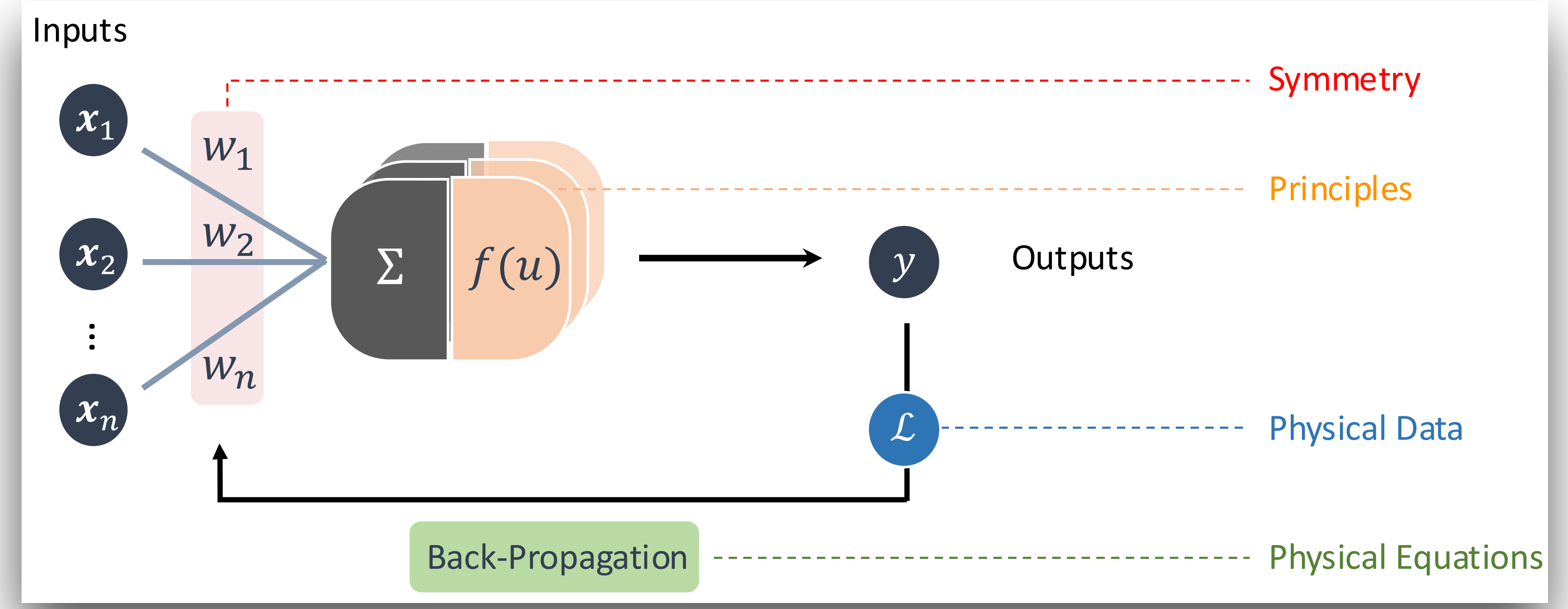


Physical data from experiments or simulations can align the model outputs with physical truths





Physics-Driven Desgins: Equations



Physical equations can be encoded into back-propagation through automatic differentiation

Wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

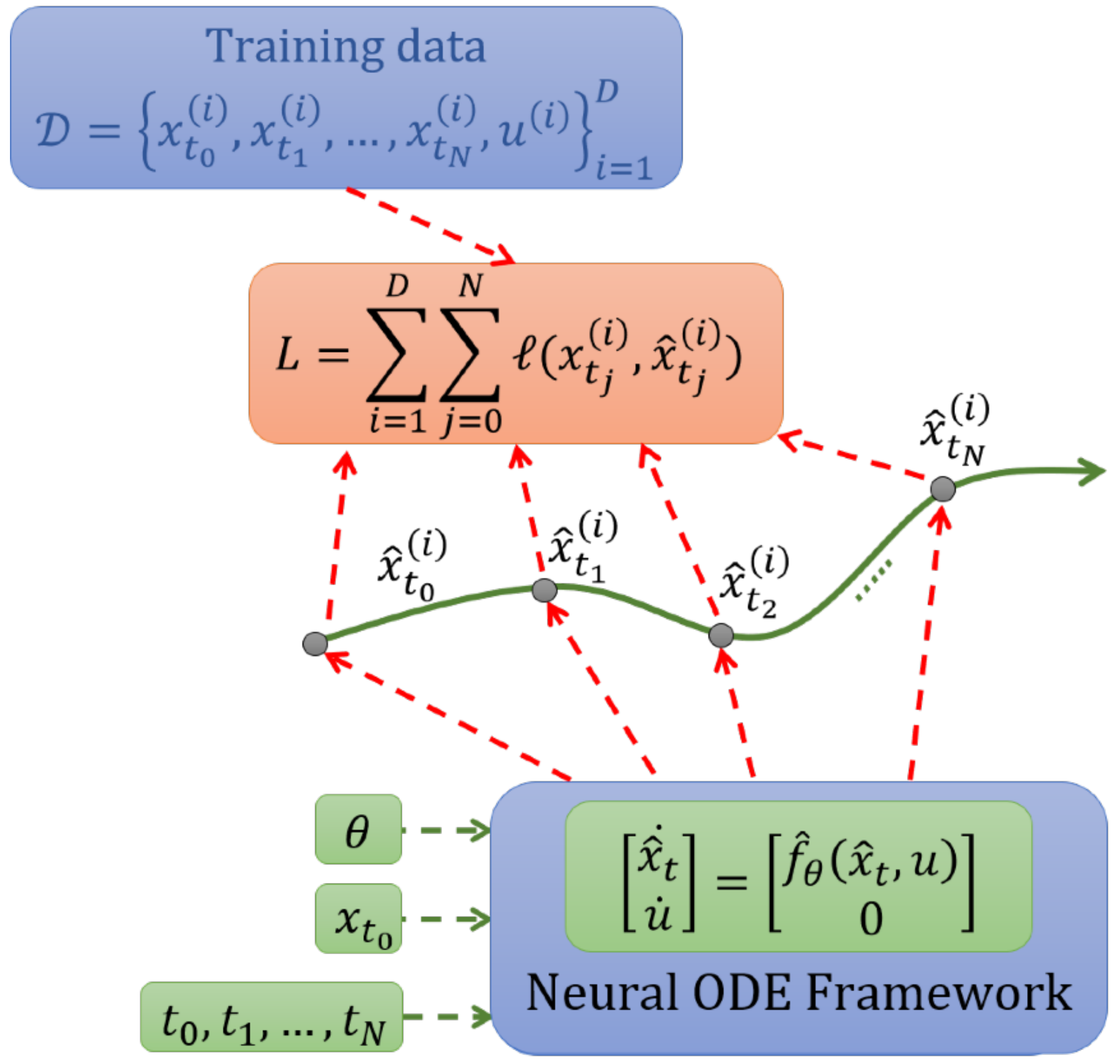
Maxwell's equations

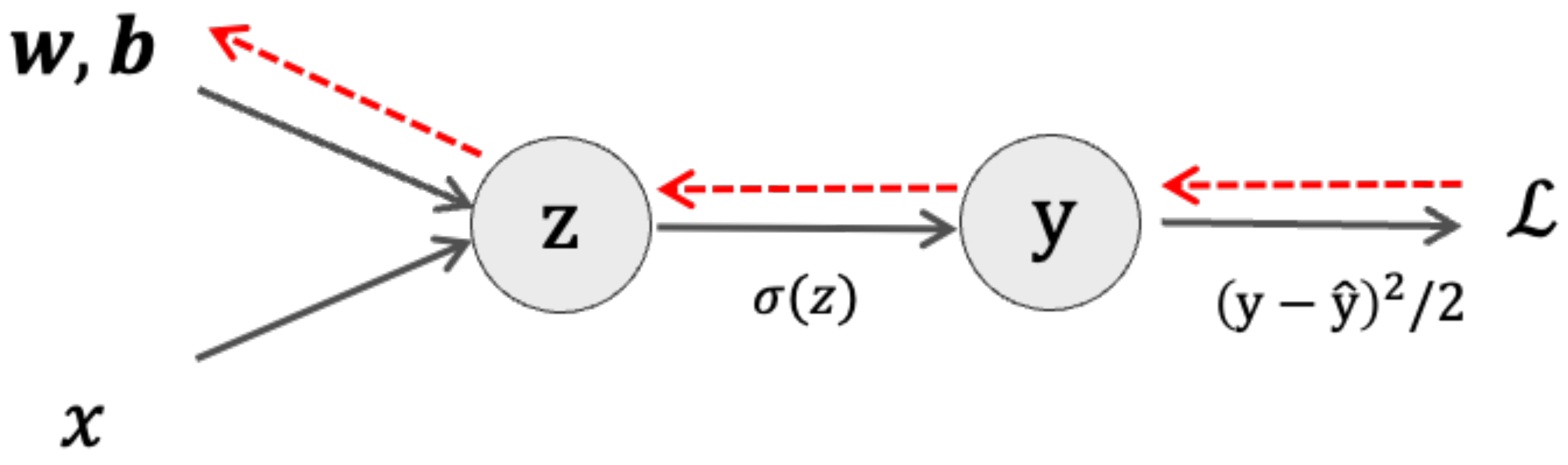
$$\nabla \cdot \mathrm{E} = 0 \qquad \nabla \times \mathrm{E} = -\frac{1}{c}\frac{\partial H}{\partial t}$$

$$\nabla \cdot \mathrm{H} = 0 \qquad \nabla \times \mathrm{H} = \frac{1}{c}\frac{\partial \mathrm{E}}{\partial t}$$

Schrödinger's equation

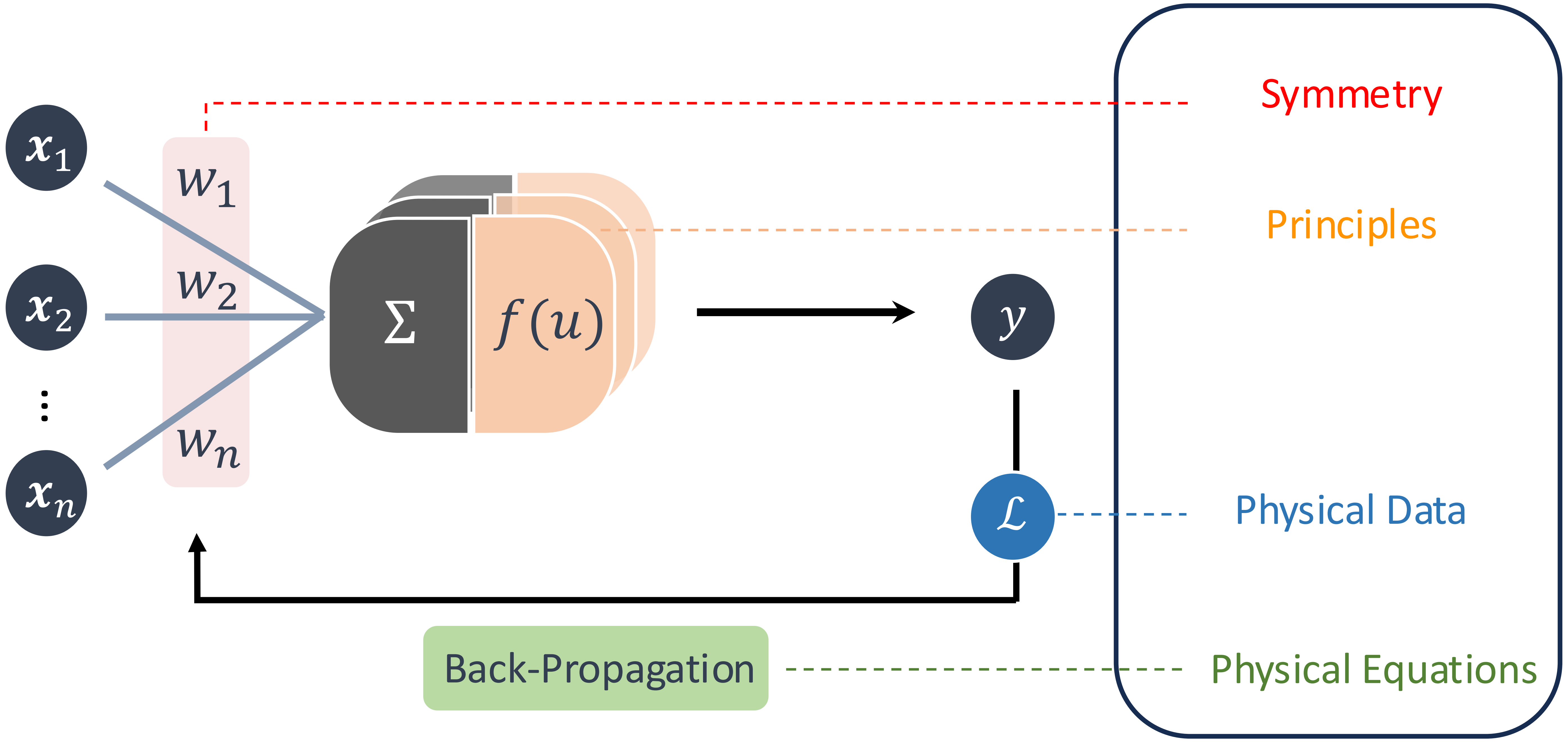
$$i\hbar \frac{\partial}{\partial t}\Psi = \hat{\mathrm{H}}\Psi$$



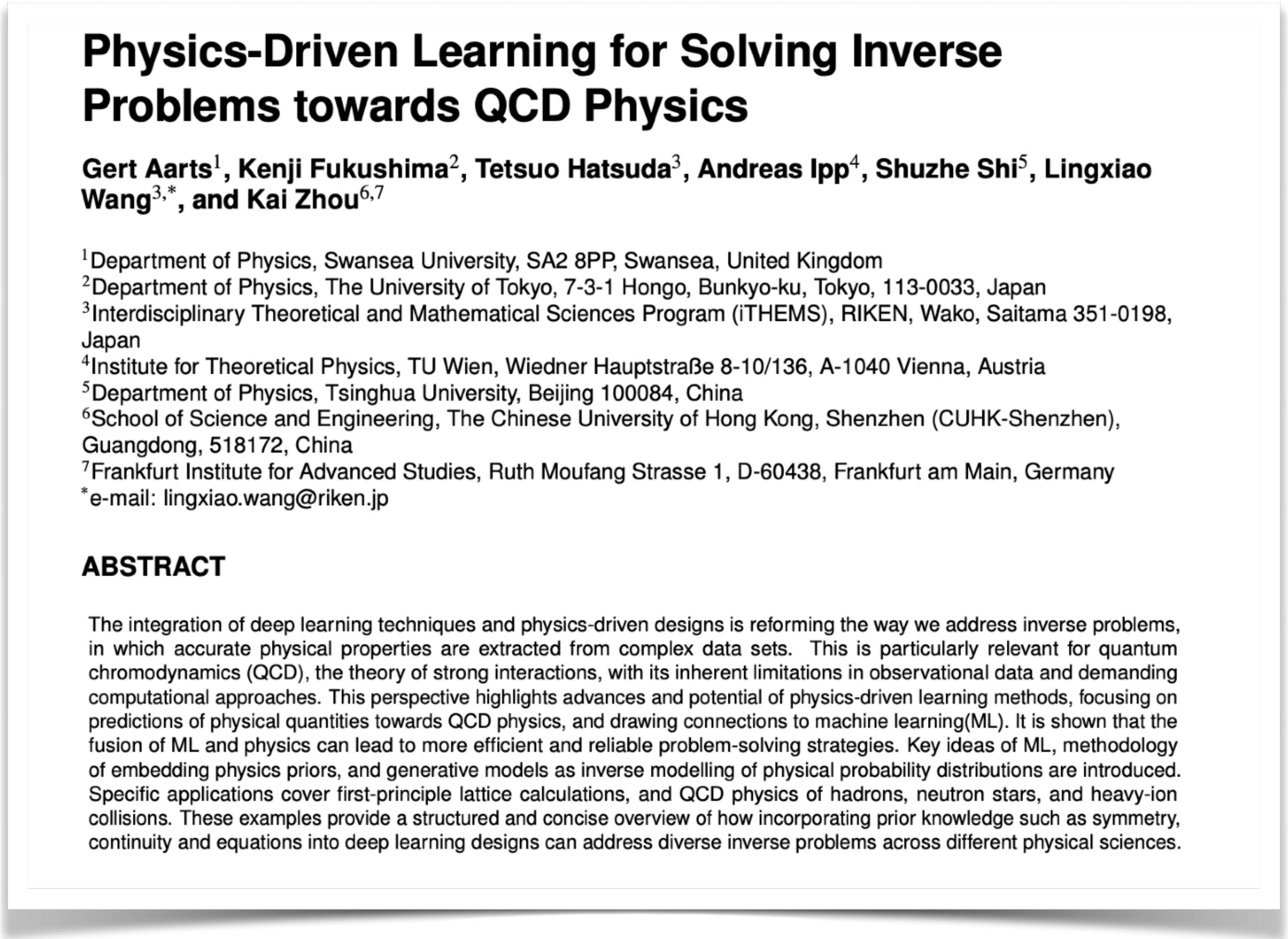


Easy-To-Compute on GPUs

Physics-Driven Desgins



accepted at *Nature Reviews Physics*



Physics-Driven Learning for Solving Inverse Problems towards QCD Physics

Gert Aarts[1], Kenji Fukushima[2], Tetsuo Hatsuda[3], Andreas Ipp[4], Shuzhe Shi[5], Lingxiao Wang[3,*], and Kai Zhou[6,7]

[1]Department of Physics, Swansea University, SA2 8PP, Swansea, United Kingdom
[2]Department of Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
[3]Interdisciplinary Theoretical and Mathematical Sciences Program (iTHEMS), RIKEN, Wako, Saitama 351-0198, Japan
[4]Institute for Theoretical Physics, TU Wien, Wiedner Hauptstraße 8-10/136, A-1040 Vienna, Austria
[5]Department of Physics, Tsinghua University, Beijing 100084, China
[6]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), Guangdong, 518172, China
[7]Frankfurt Institute for Advanced Studies, Ruth Moufang Strasse 1, D-60438, Frankfurt am Main, Germany
*e-mail: lingxiao.wang@riken.jp

ABSTRACT

The integration of deep learning techniques and physics-driven designs is reforming the way we address inverse problems, in which accurate physical properties are extracted from complex data sets. This is particularly relevant for quantum chromodynamics (QCD), the theory of strong interactions, with its inherent limitations in observational data and demanding computational approaches. This perspective highlights advances and potential of physics-driven learning methods, focusing on predictions of physical quantities towards QCD physics, and drawing connections to machine learning(ML). It is shown that the fusion of ML and physics can lead to more efficient and reliable problem-solving strategies. Key ideas of ML, methodology of embedding physics priors, and generative models as inverse modelling of physical probability distributions are introduced. Specific applications cover first-principle lattice calculations, and QCD physics of hadrons, neutron stars, and heavy-ion collisions. These examples provide a structured and concise overview of how incorporating prior knowledge such as symmetry, continuity and equations into deep learning designs can address diverse inverse problems across different physical sciences.

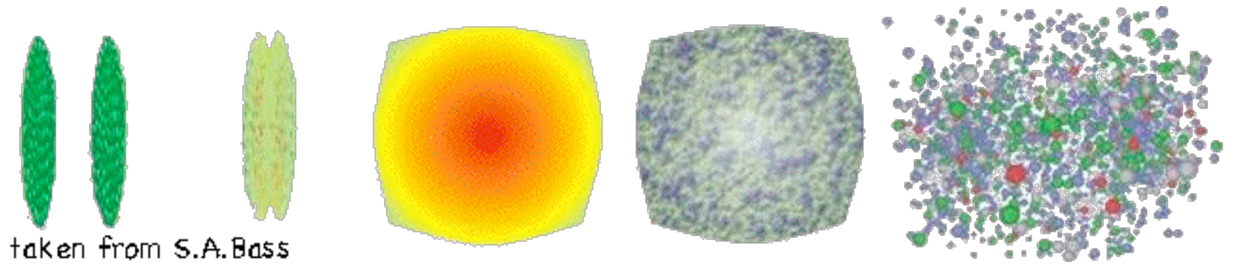
Physics Knowledge for Designing
More Controllable and Reliable Deep Models

Inverse Problems in QCD Physics

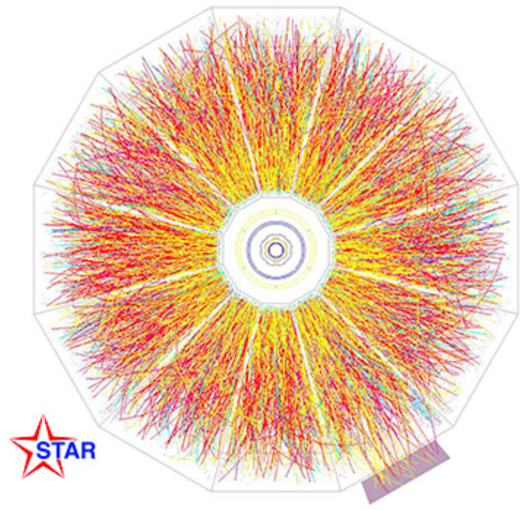
Inverse Problems

Physics

Data

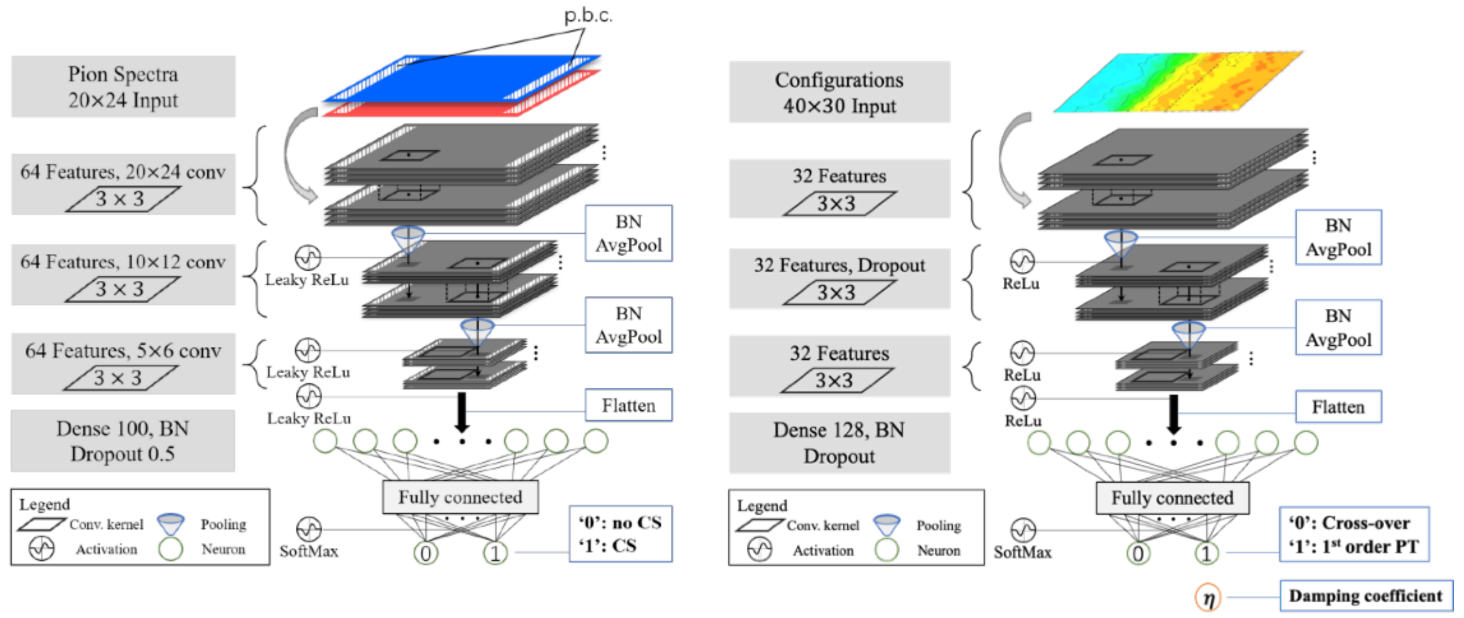


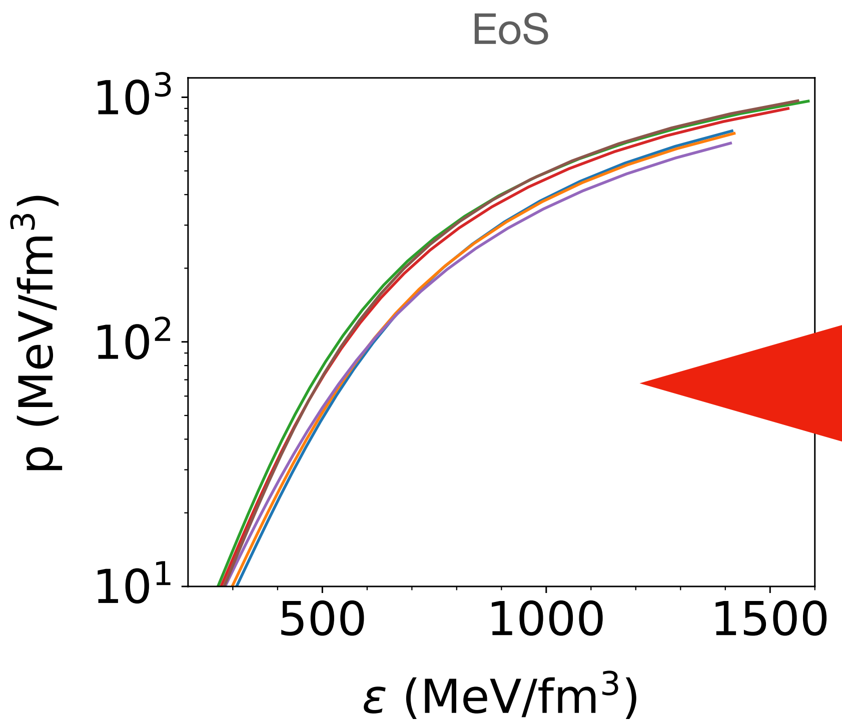
QCD matter properties

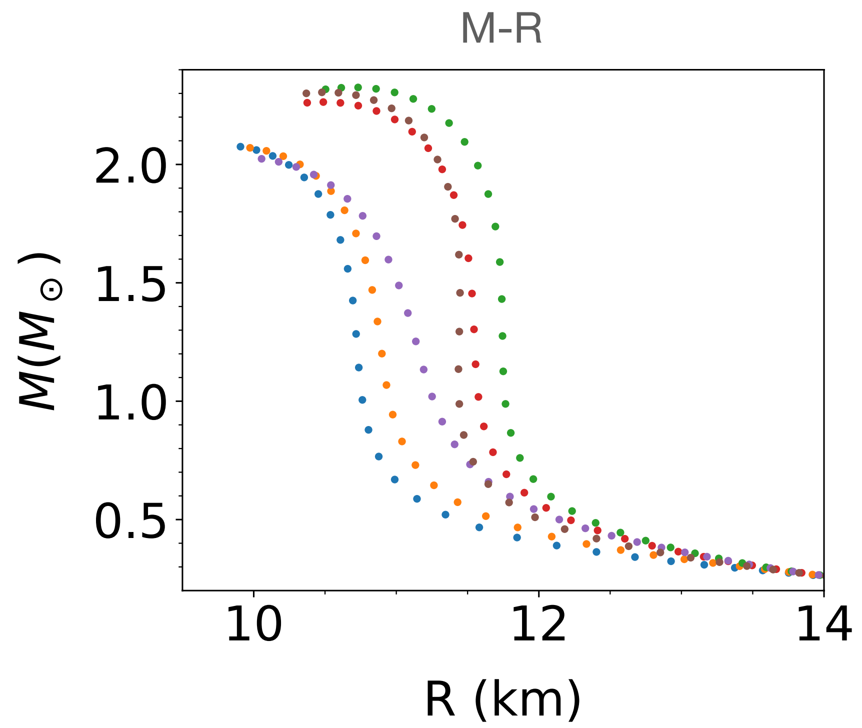


Heavy-Ion Collisions

Phys. Rev. C 106, L051901; Phys. Rev. D 103, 116023

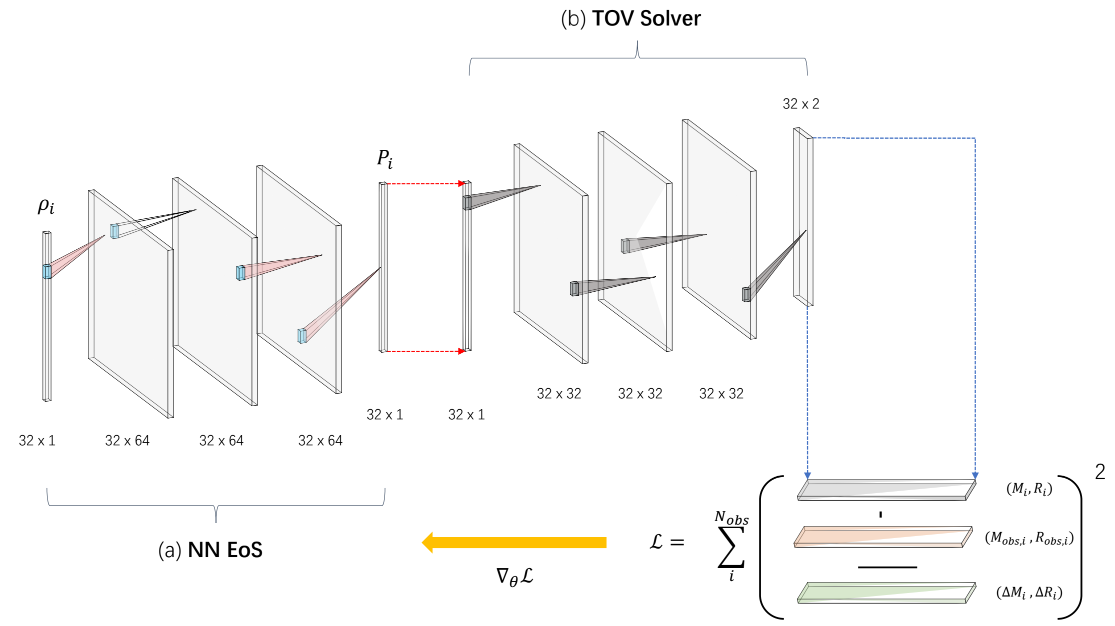


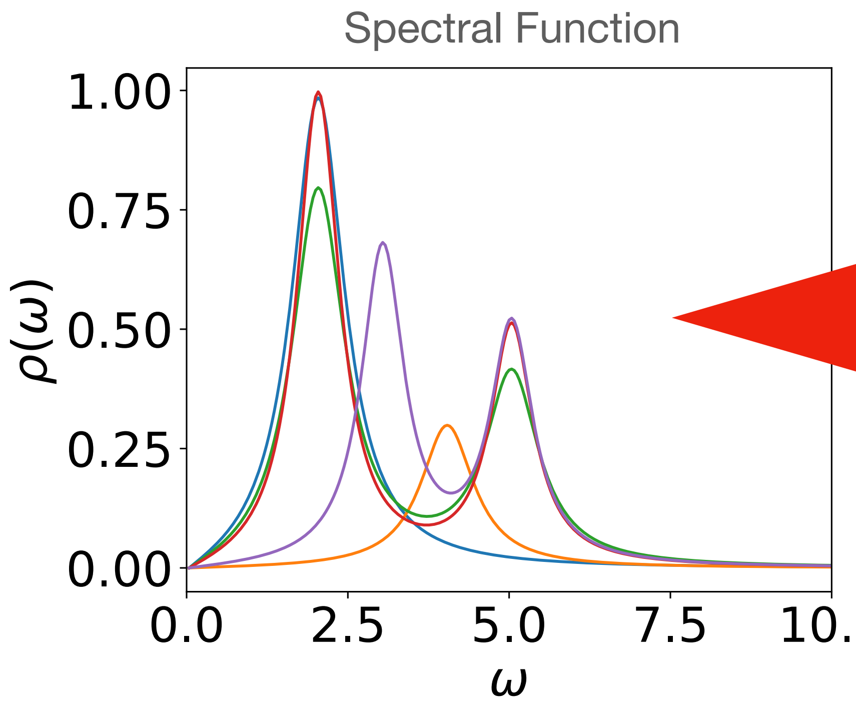


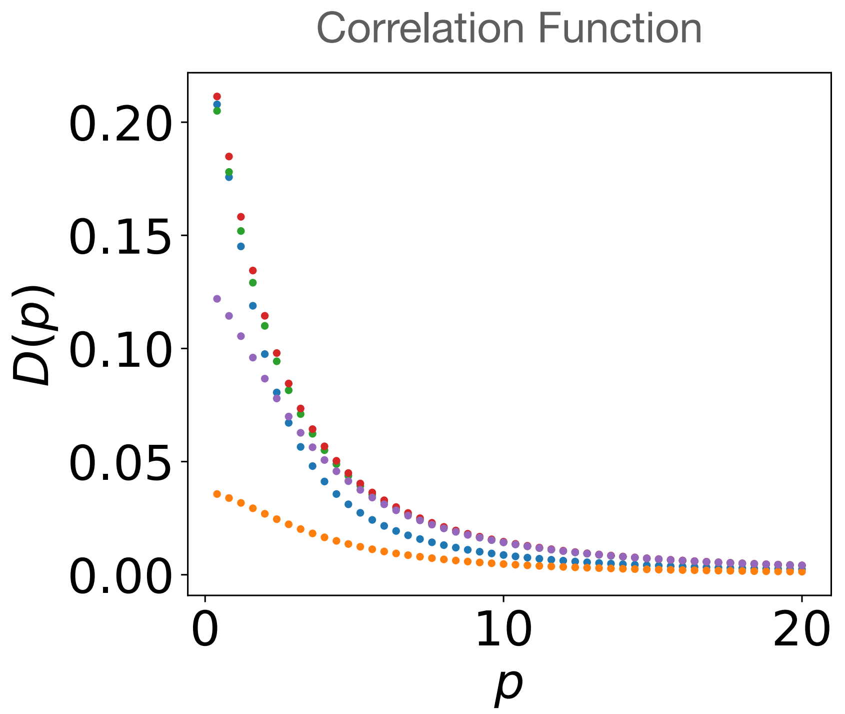


Neutron Star

Phys. Rev. D 107, 083028; JCAP08 (2022) 071

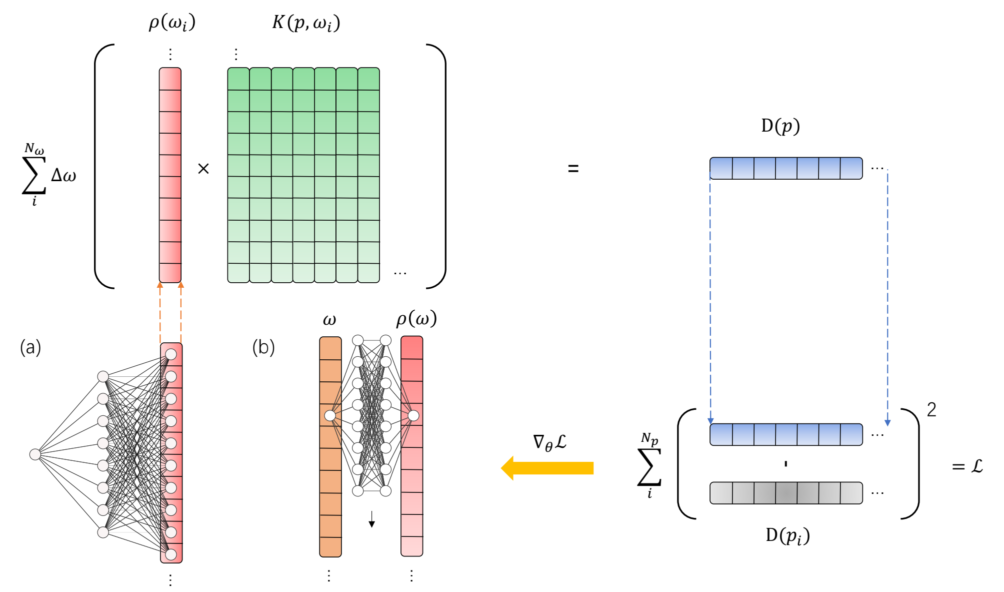






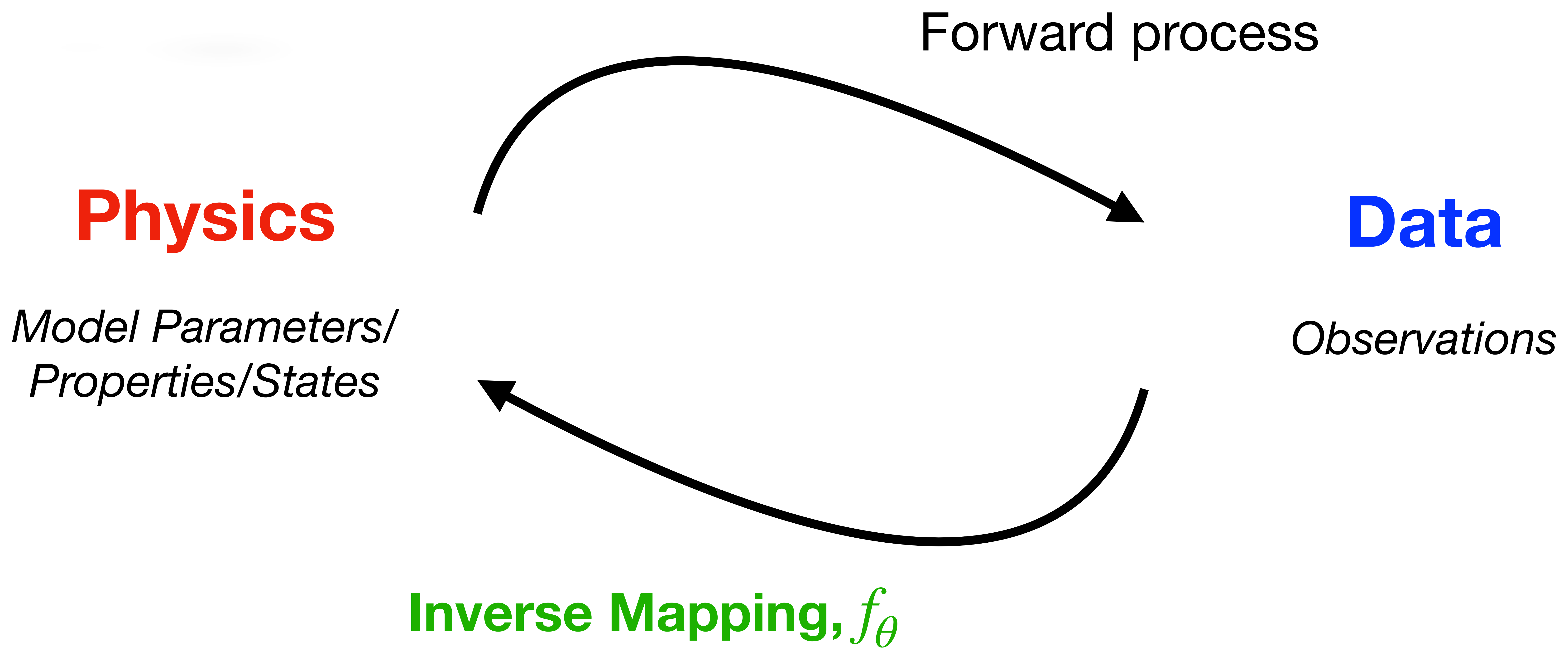
Lattice

Phys. Rev. D 106, L051502; Comput. Phys. Commun. 282, 108547 (2023);



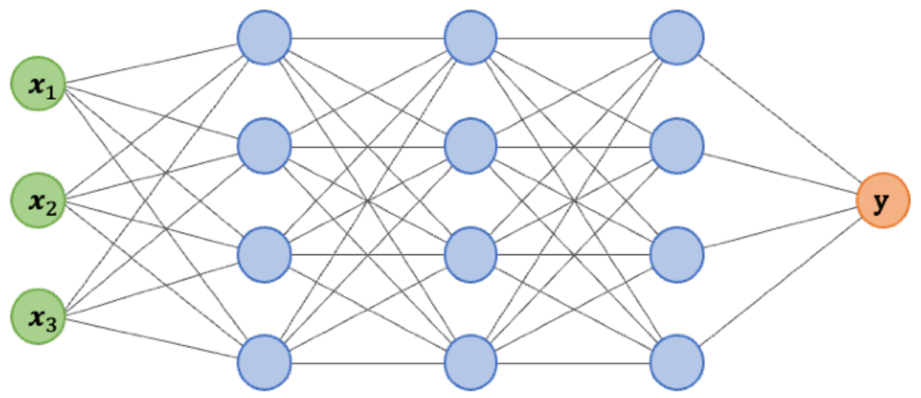
Data-Driven Learning

$$f_\theta : X \to Y$$

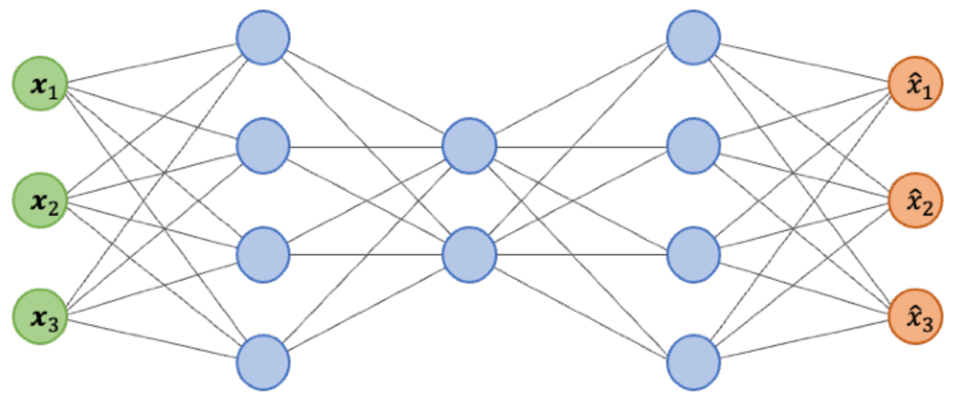


Data-Driven Learning

$$f_\theta : X \rightarrow Y$$



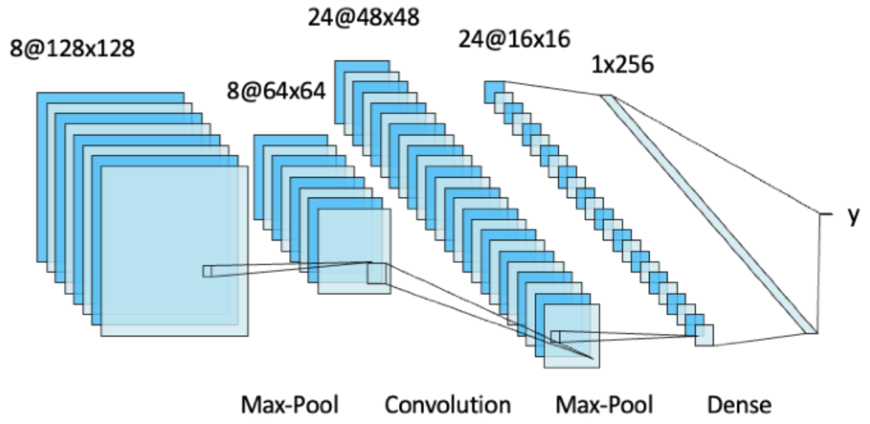
Deep Neural Network



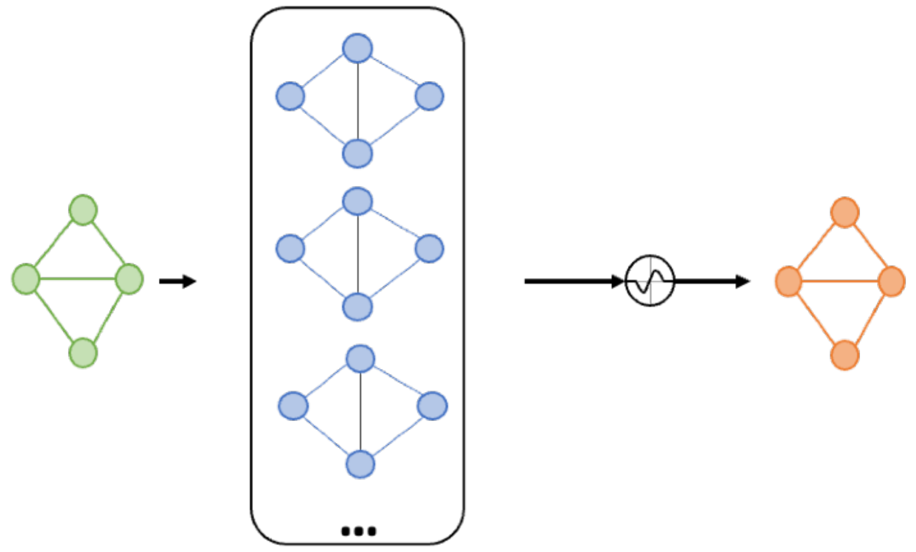
AutoEncoder

Universal Approximation Theorem (1989,1991)

A feed-forward network with *a single hidden layer* containing *a finite number of neurons* can approximate arbitrary continuous functions.



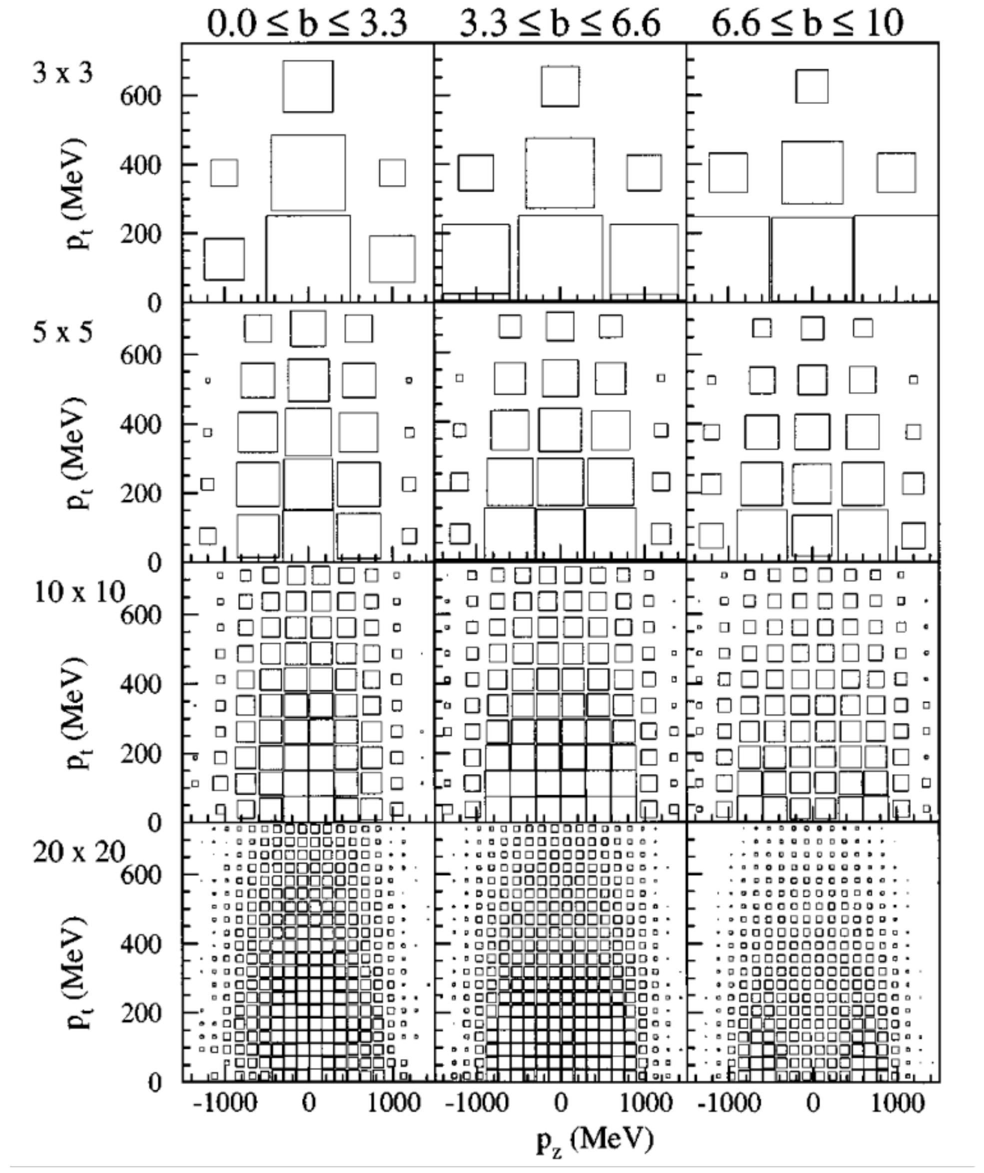
Convolutional Neural Network

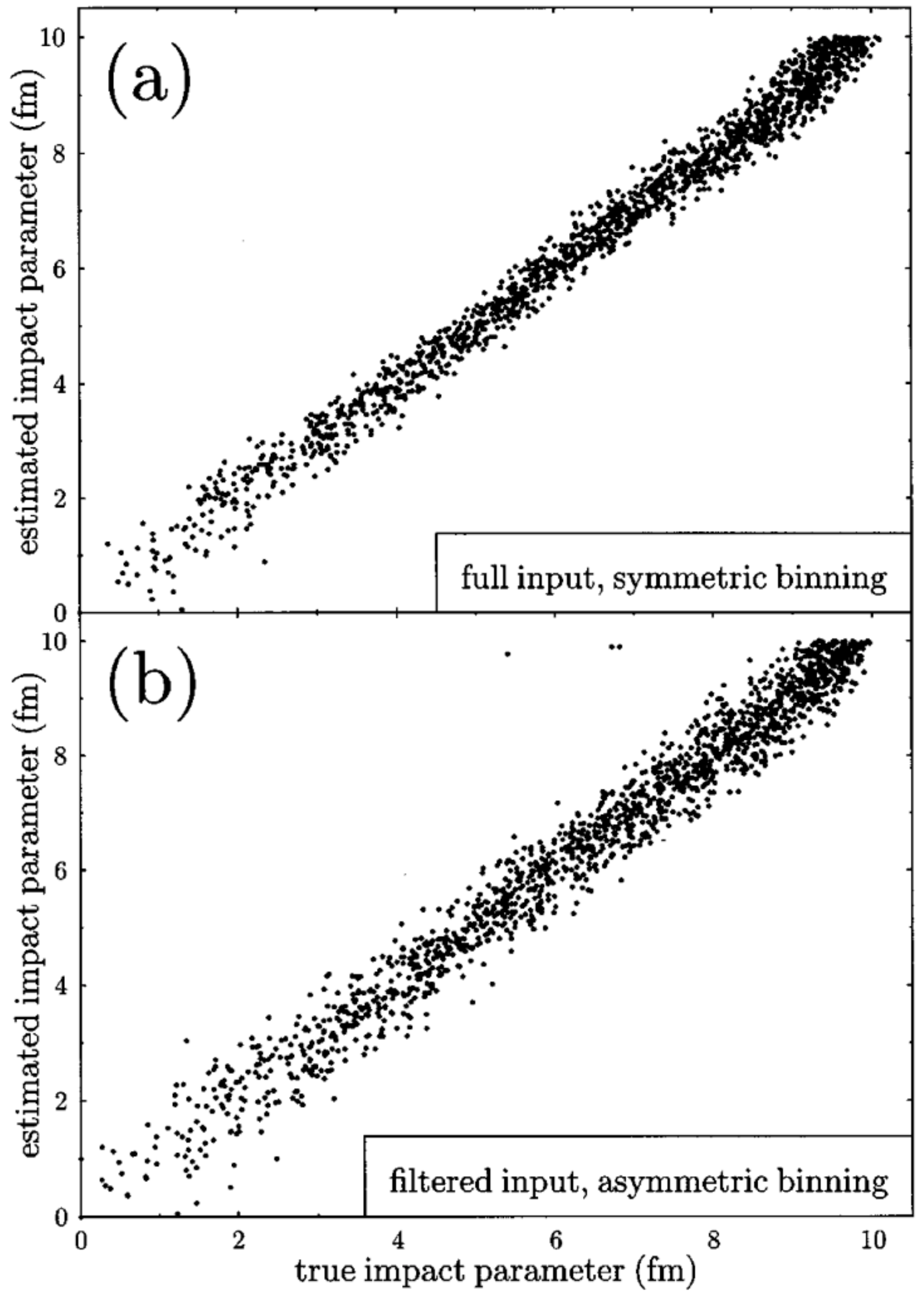


Graph Neural Network

Data-Driven Learning

Determine Impact Parameter





S. A. Bass, A. Bischoff, J. A. Maruhn, H. Stöcker, and W. Greiner, Phys. Rev. C 53, 2358 (1996)

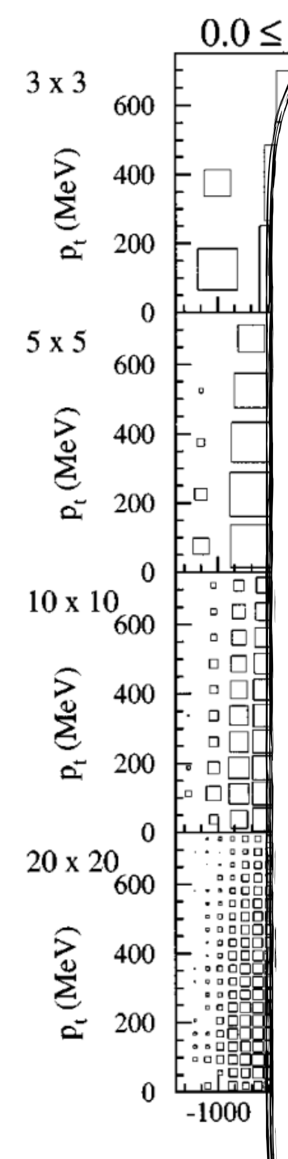
QMC data

1 hidden layer with 20 neurons

Input 5X5

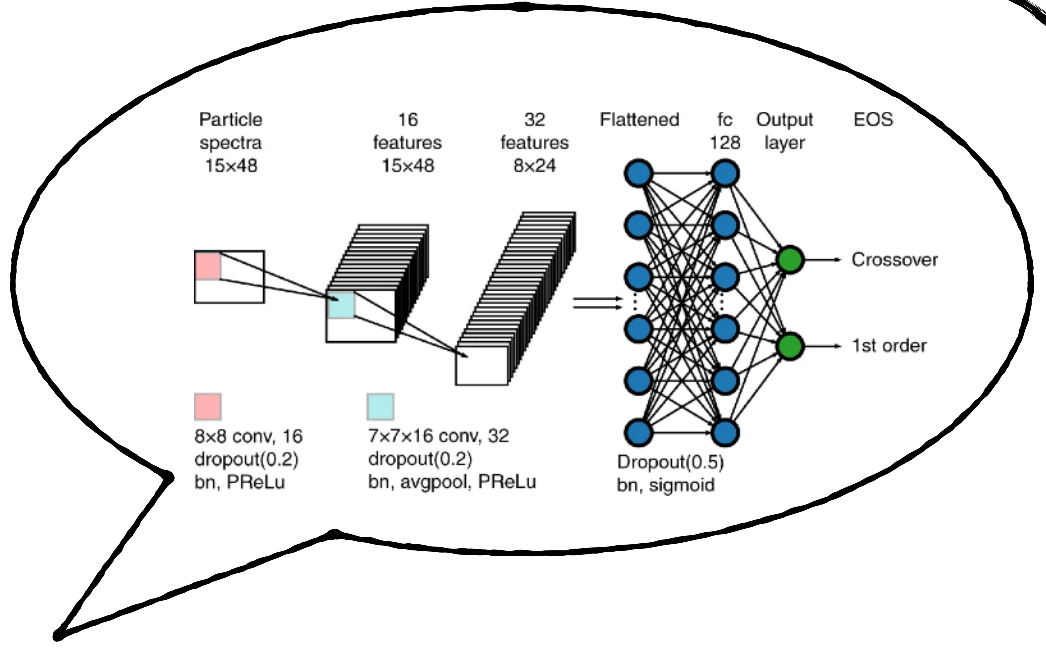
Data-Driven Learning

Determine Impact Parameter

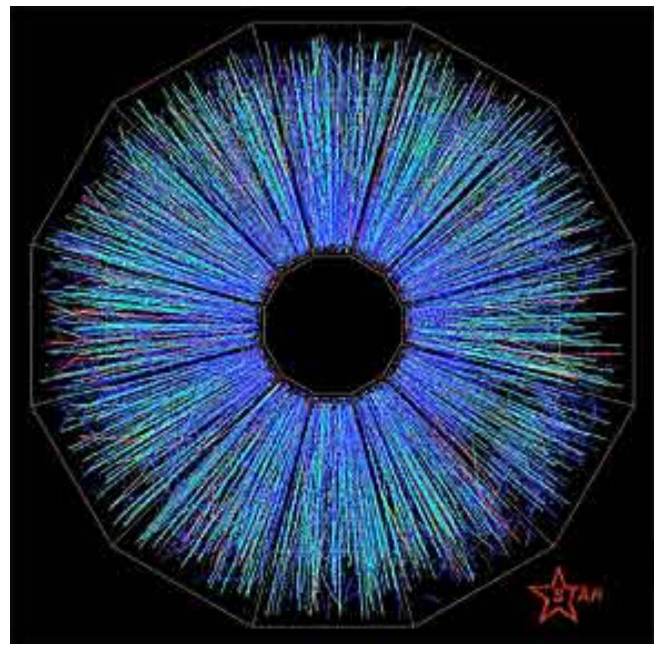


S. A

Recognizing QCD Phase Transitions



$\rho(p_T, \Phi)$



→

→

Cross-Over
or
1st order PT

L.-G. Pang, K. Zhou, N. Su, H. Petersen, H. Stöcker, and X.-N. Wang, Nature Commun. 9, 210 (2018)

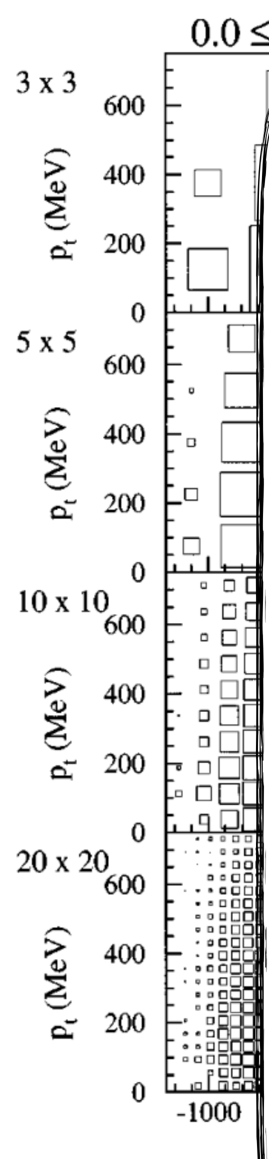
Hydro data

CNNs+DNNs

Input 15X48

Data-Driven Learning

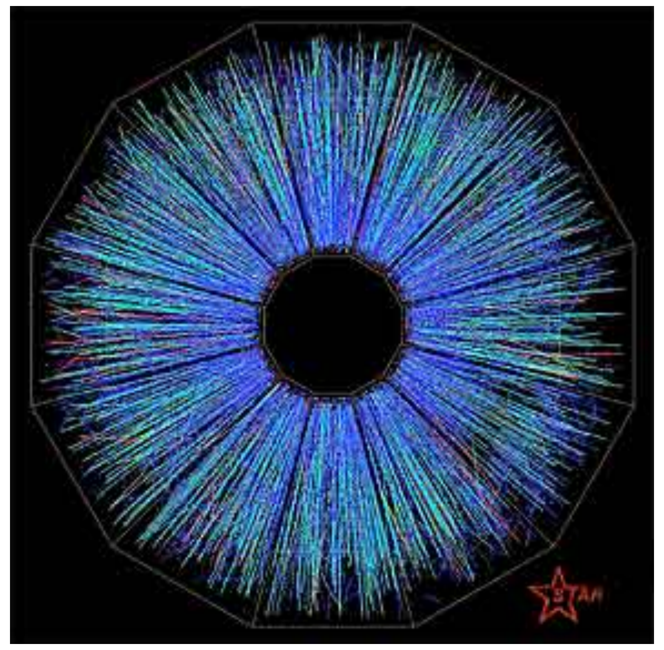
Determine Impact Parameter



S. A

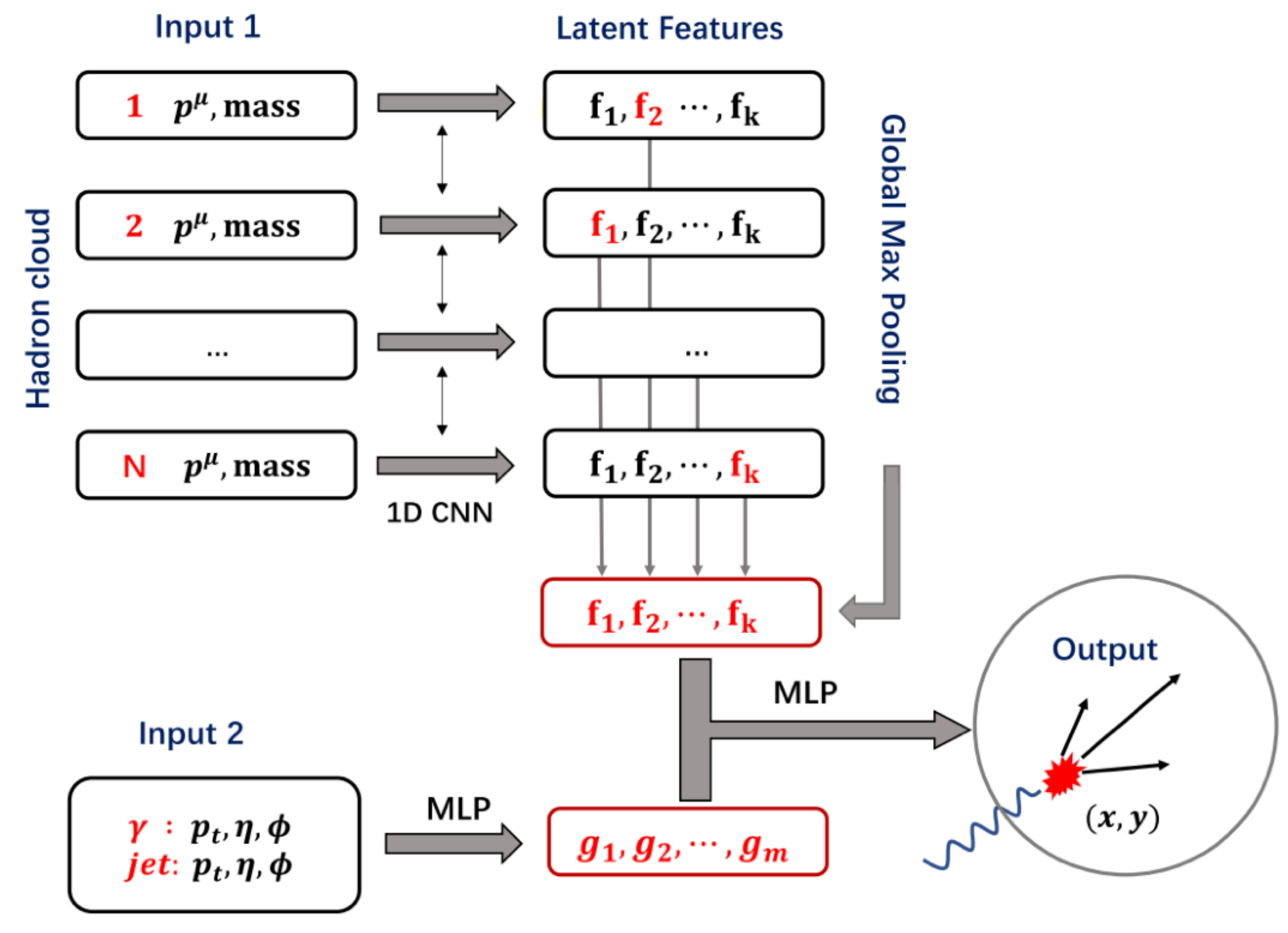
Recognizing QCD Phase Transitions

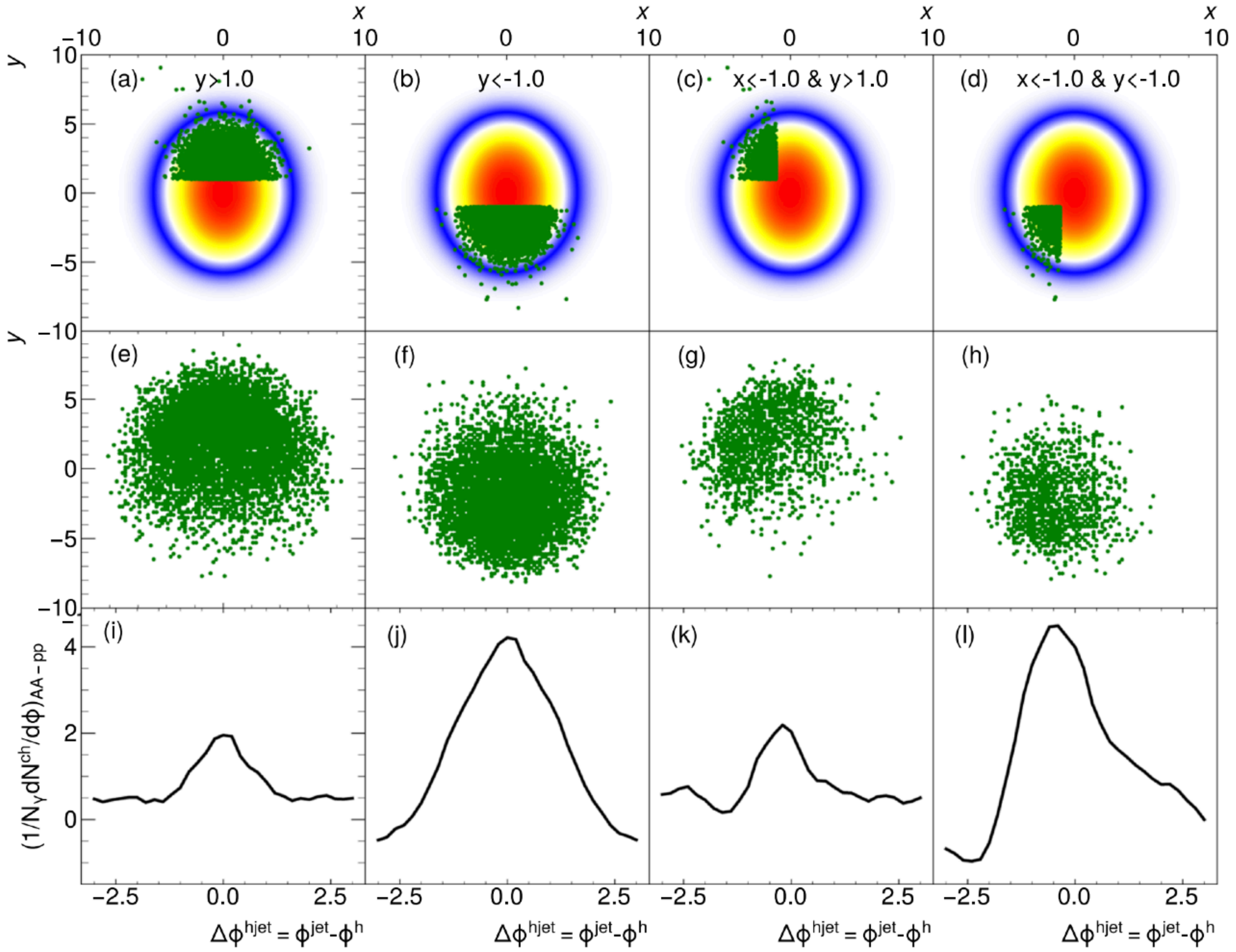
$\rho(p_T, \Phi)$



L.-G. Pang, K. Zhou, N. S

Point Net for Jet Tomography

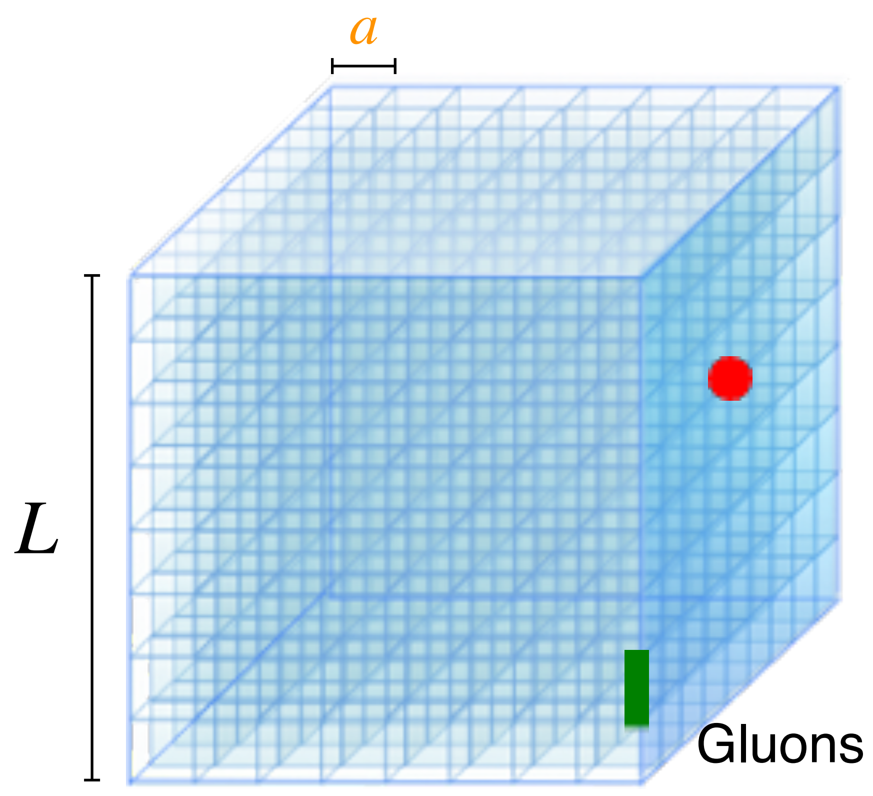




Z. Yang, Y. He, W. Chen, W.-Y. Ke, L.-G. Pang, and X.-N. Wang, Eur. Phys. J. C 83, 652 (2023).

Hadron Interactions

$$\mathscr{L} = -\frac{1}{4}G^a_{\mu\nu}G^{\mu\nu}_a + \bar{q}\gamma^\mu(i\partial_\mu - gt^aA^a_\mu)q - m\bar{q}q$$



4-d Euclidean Lattice

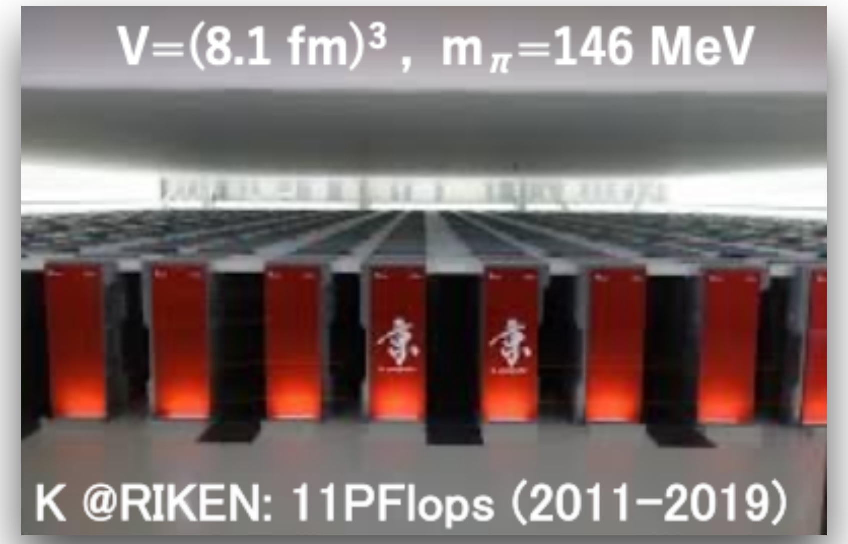
Quarks on sites $q(x)$

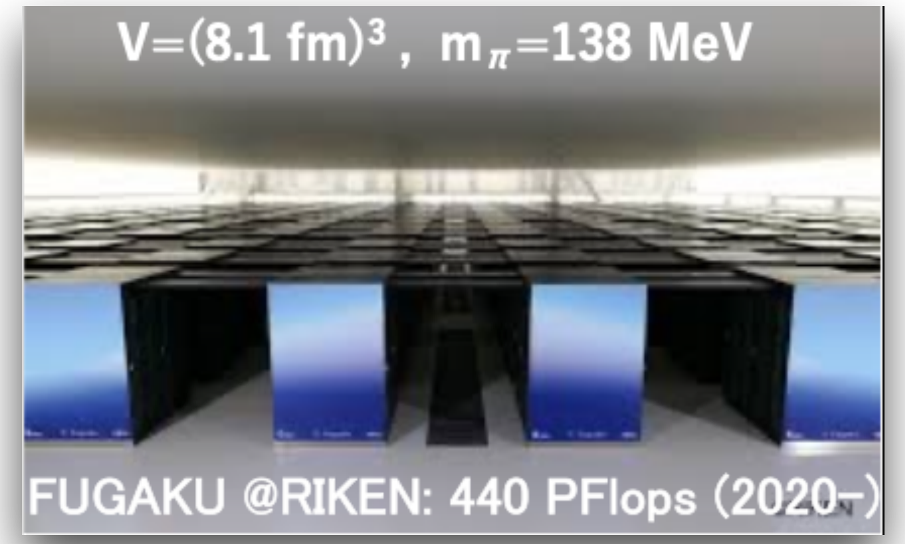
Gluons on links $U_\mu(x, x+\mu) = \exp[-iaA_\mu]$

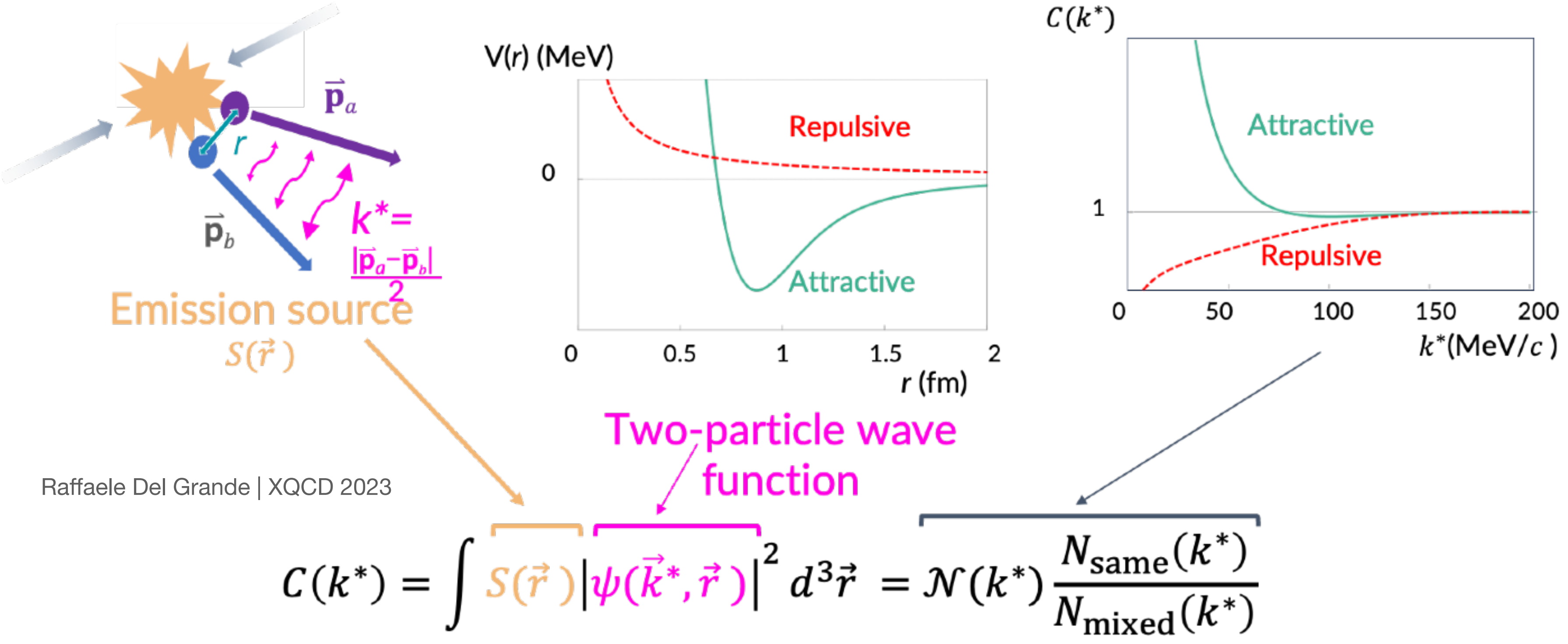
Huge integration variables
$\sim 10^{9-10}$ for 96^4 lattice, ~ 50 GB/config

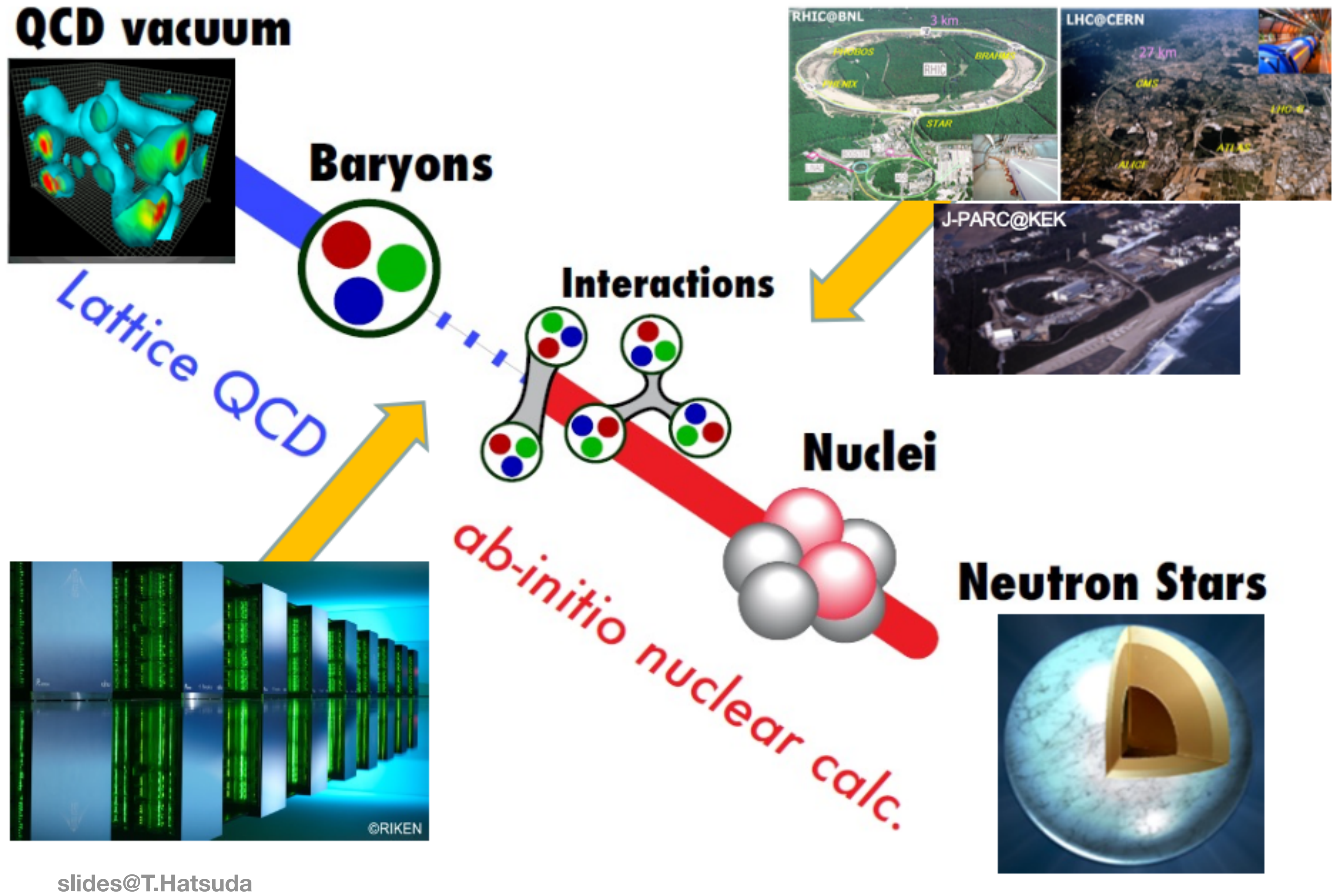
Importance Sampling
Hybrid MC = MD + Metropolis

Continuum & Thermodynamic Limits
$a \to 0, L \to \infty$



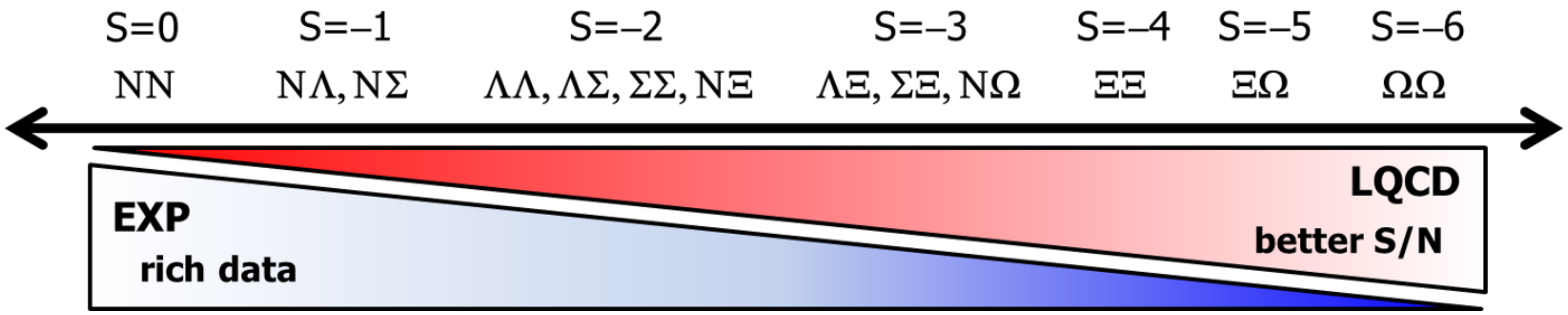




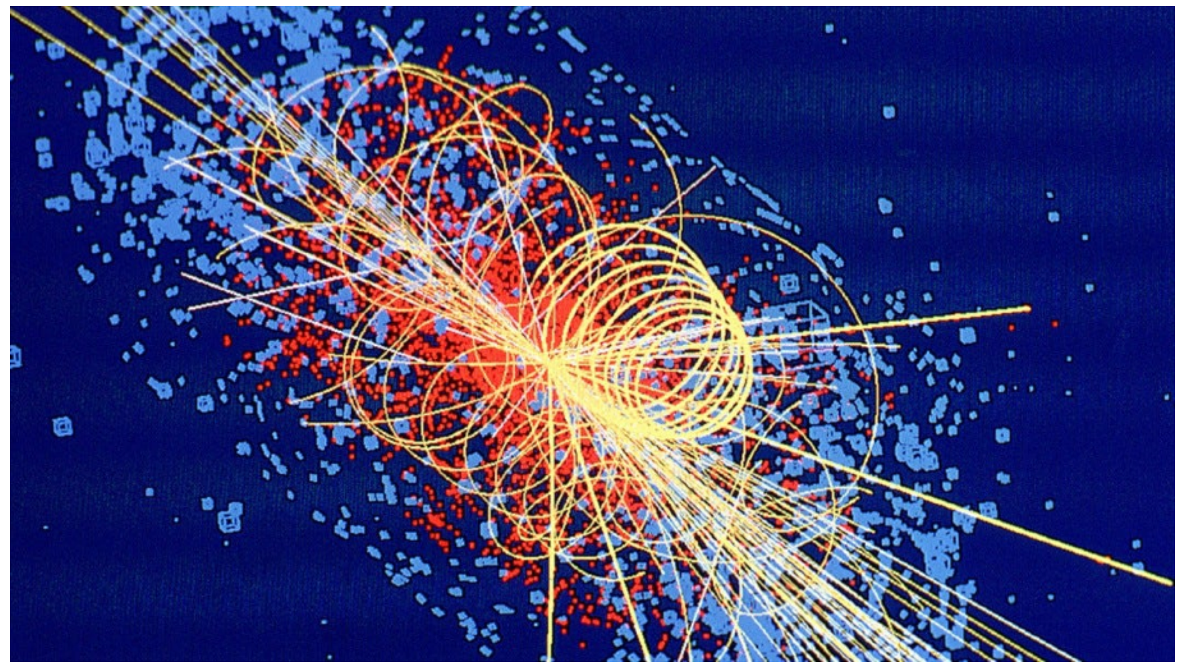


slides@T.Hatsuda

Hadron Interactions



LHC@CERN, RHIC@BNL
J-PARC@KEK, FAIR@GSI, HIAF



Femtoscopy
HAL QCD method

+

Deep Learning

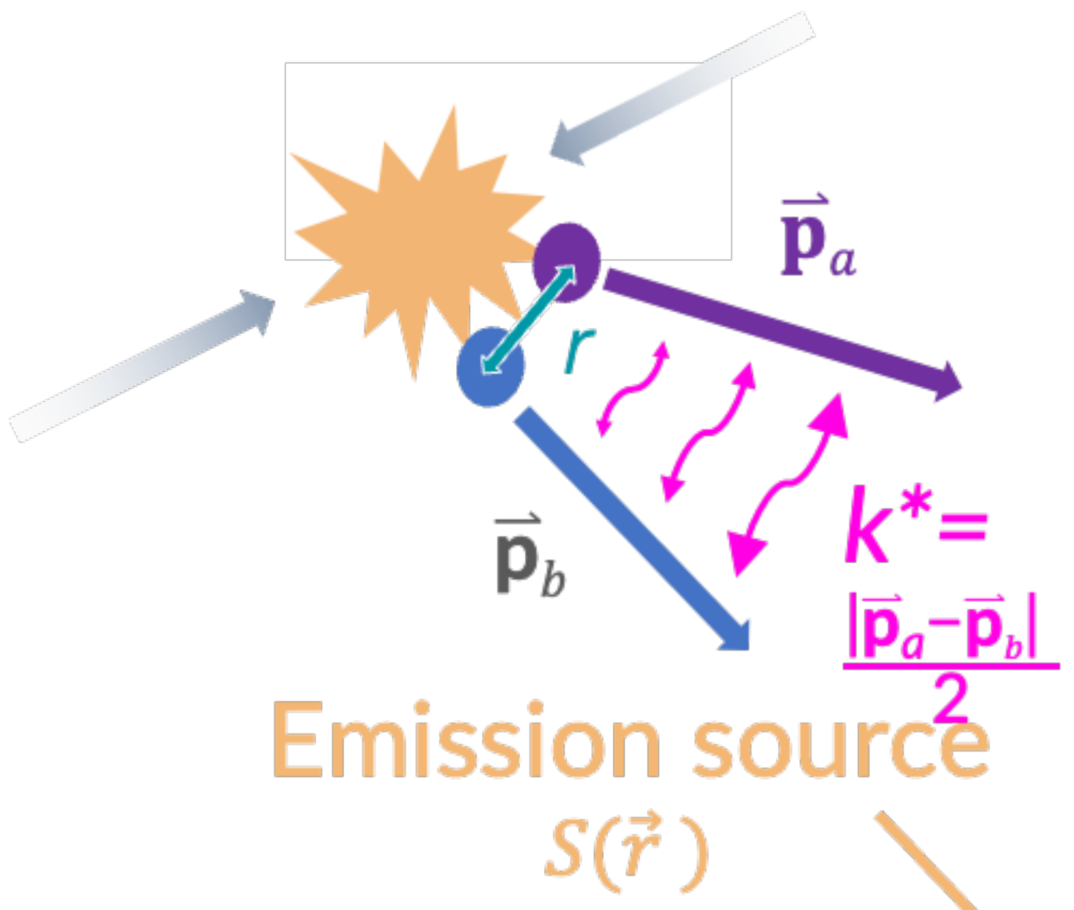
↓

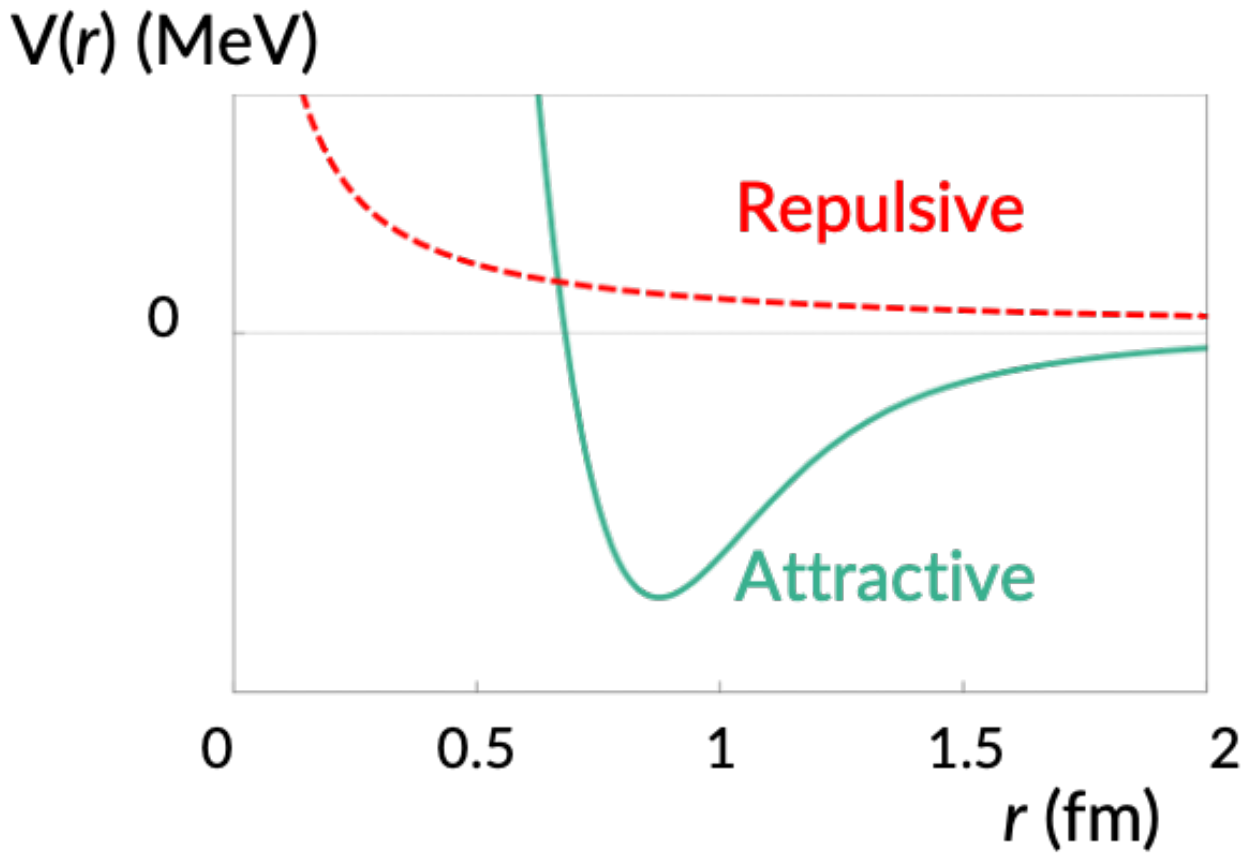
?

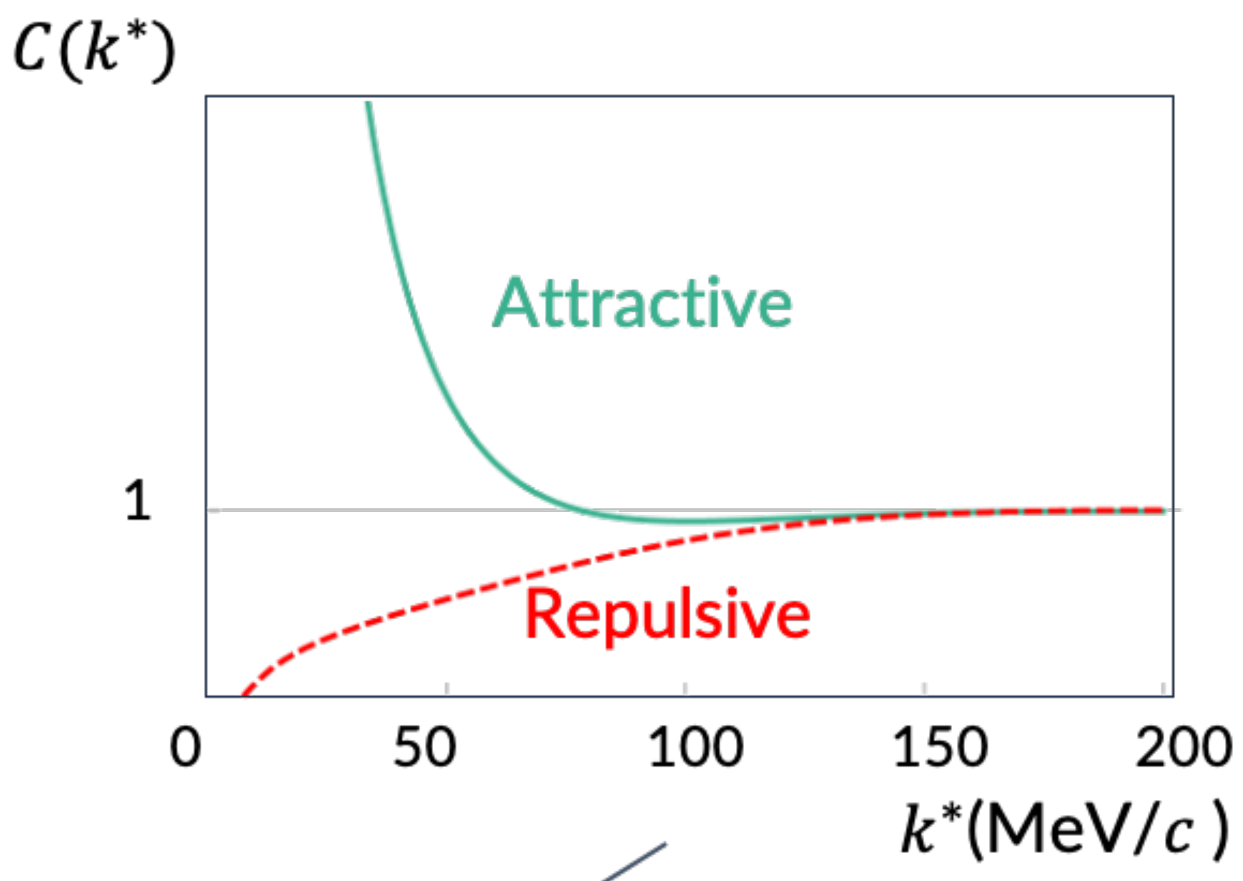
Hadrons to Atomic nuclei from Lattice QCD
(HAL QCD Collaboration)

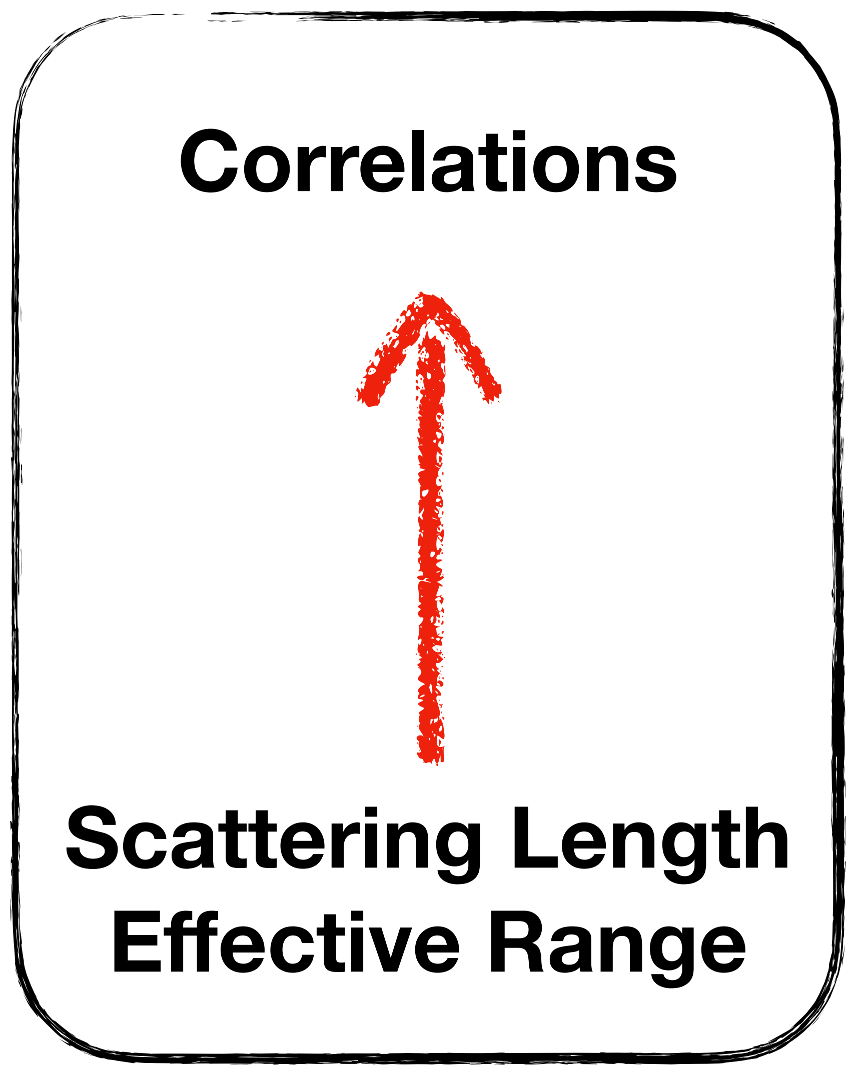
S. Aoki, T. M. Doi, E. Itou (Kyoto U.) **T. Aoyama** (ISSP)
T. Doi, T. Hatsuda, Y. Lyu, L. Wang, R. Yamada, L. Zhang (RIKEN)
Y. Ikeda, N. Ishii, P. Junnarkar, H. Nemura, K. Sasaki(Osaka U.)
T. Inoue (Nihon U.) **K. Murakami** (TITech)
K. Murase (Tokyo Metropolitan U.) **F. Etminan** (U. of Birjand)
T. Sugiura (Rissho U.) **H. Tong** (U. of Bonn)

Femtoscopy









Two-particle wave function

$$C(k^*) = \int S(\vec{r}\,)\left|\psi(\vec{k}^*, \vec{r}\,)\right|^2 d^3\vec{r} \;=\; \mathcal{N}(k^*)\,\frac{N_{\mathrm{same}}(k^*)}{N_{\mathrm{mixed}}(k^*)}$$

Raffaele Del Grande | XQCD 2023

Lednicky-Lyuboshits(LL) analytic model
(Asymptotic wave-function+
Effective range correlation+
Gaussian source)

Lednicky, Lyuboshits, Sov.J.Nucl.Phys. 35 (1982) 770

Femtoscopy

Asymptotic wave-function

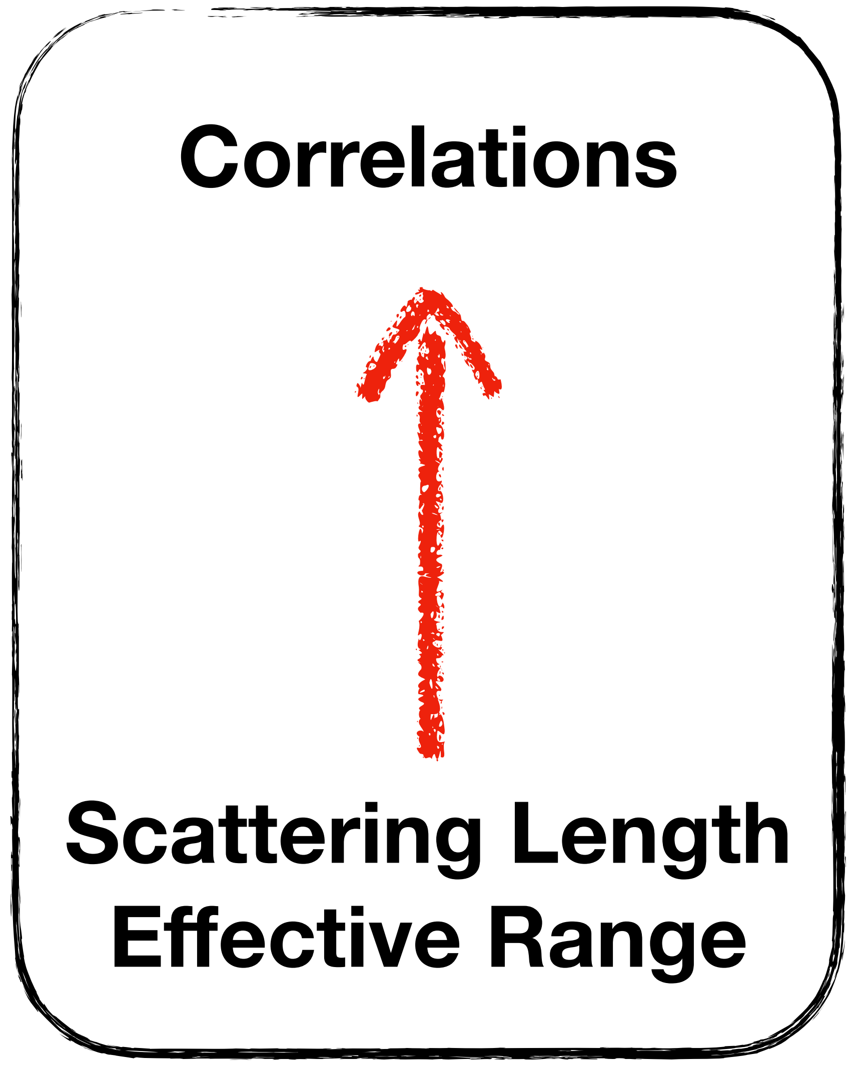
$$\psi_0(r) \to \psi_{\text{asy}}(r) = \frac{e^{-i\delta}}{qr}\sin(qr+\delta) = \mathcal{S}^{-1}\left[\frac{\sin qr}{qr} + f(q)\frac{e^{iqr}}{r}\right]$$

$$\begin{aligned} C_{\text{LL}}(q) &= 1 + \int dr S_{12}(r)\left(|\psi_{\text{asy}}(r)|^2 - |j_0(qr)|^2\right) \\ &= 1 + \frac{|f(q)|^2}{2R^2}F_3\left(\frac{r_{\text{eff}}}{R}\right) + \frac{2\text{Re}f(q)}{\sqrt{\pi}R}F_1(2x) - \frac{\text{Im}f(q)}{R}F_2(2x) \end{aligned}$$

$x = qR$, R is Gaussian Size, F_1, F_2, F_3 are known functions

Scattering amplitude at low energies

$$q\cot\delta = -\frac{1}{a_0} + \frac{1}{2}r_{\text{eff}}q^2 + O(q^4) \to f(q) = (q\cot\delta - iq)^{-1}$$



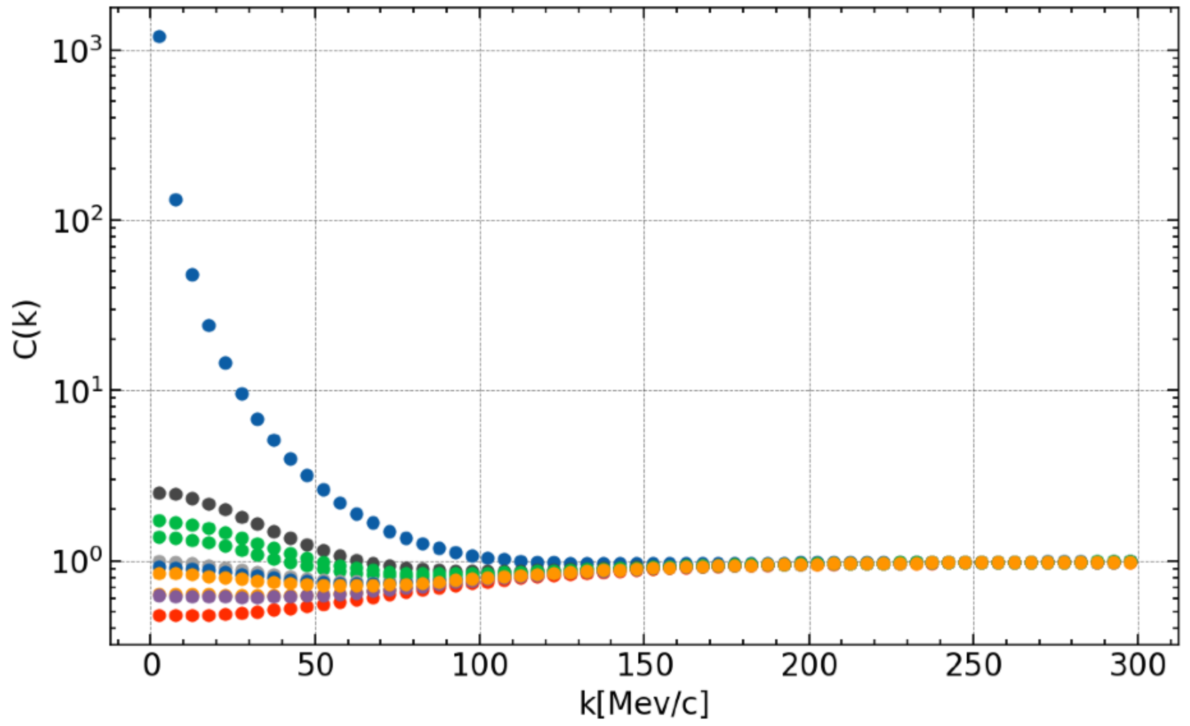
Lednicky-Lyuboshits(LL) analytic model
(Asymptotic wave-function+
Effective range correlation+
Gaussian source)

Lednicky, Lyuboshits, Sov.J.Nucl.Phys. 35 (1982) 770

Inverse Femtoscopy

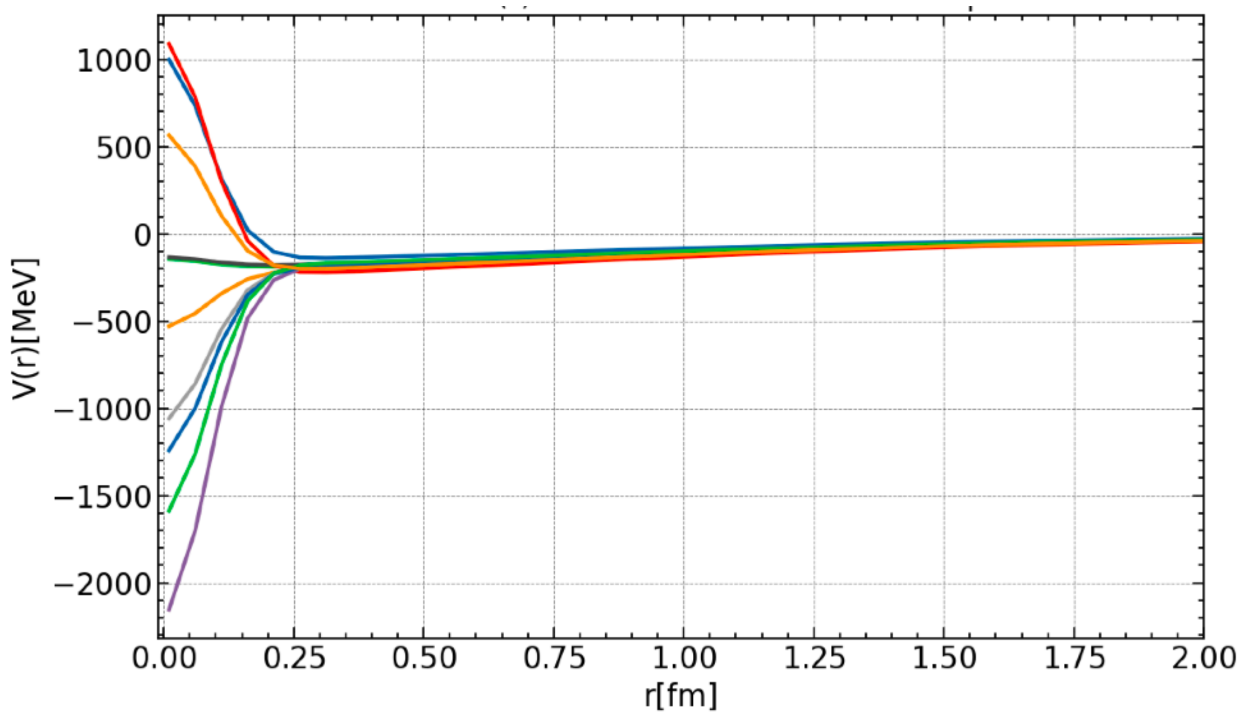
in Preparation with Jiaxing Zhao, etc.

$$C(k^*) = \int S(\vec{r}) \left| \psi(\vec{k}^*, \vec{r}) \right|^2 d^3\vec{r} = \mathcal{N}(k^*) \frac{N_{\text{same}}(k^*)}{N_{\text{mixed}}(k^*)}$$



Does this inverse mapping exist?

$$V(r)$$



Correlations

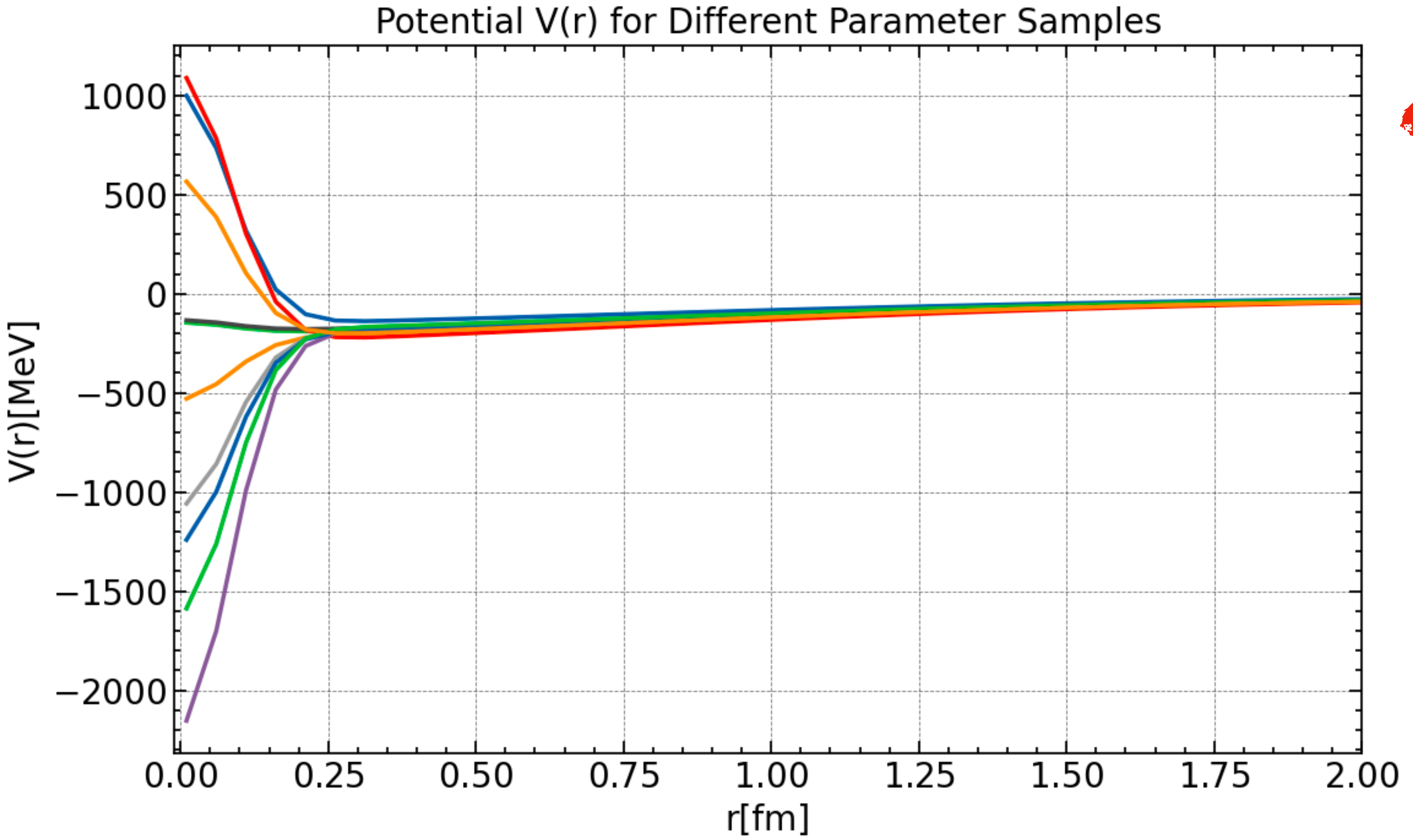
↓ ?

Potentials

Inverse Femtoscopy

Potential Functions

$$V(r) = b_1 e^{-b_2 r^2} + b_3(1 - e^{-b_4 r^2})(\frac{e^{(-m_\pi r)}}{r})^{n_\pi}$$



Source Function

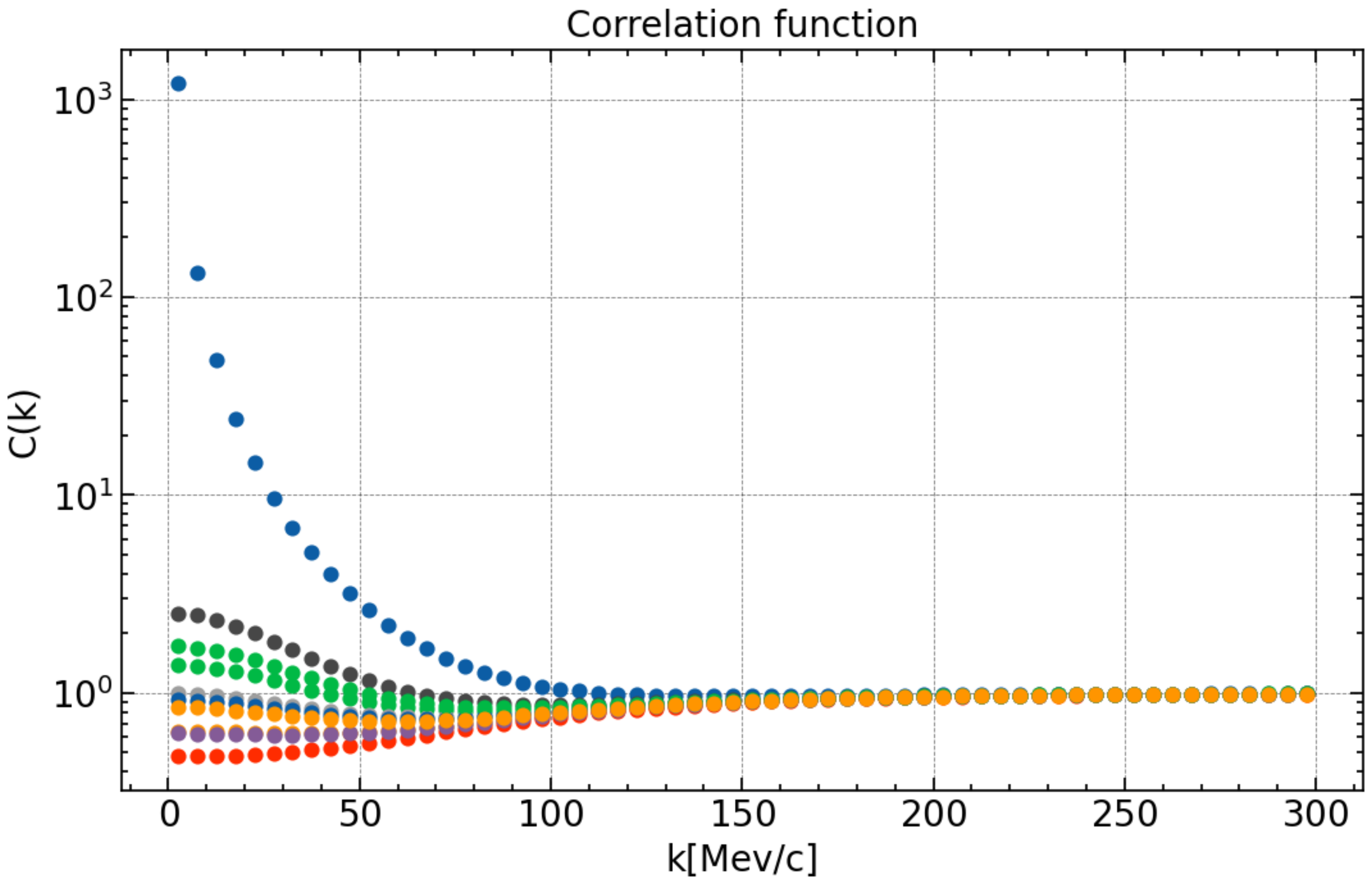
$$S(r) = (4\pi r_0^2)^{-3/2} e^{-\frac{r^2}{4r_0^2}} \qquad r_0 = 1.3\ \textbf{fm}$$

Deep Neural Network (DNN)

Schrödinger eq.

CATS Framework: D. Mihaylov et al., Eur. Phys. J. C78 (2018) 394

in Preparation **with Jiaxing Zhao, etc.**

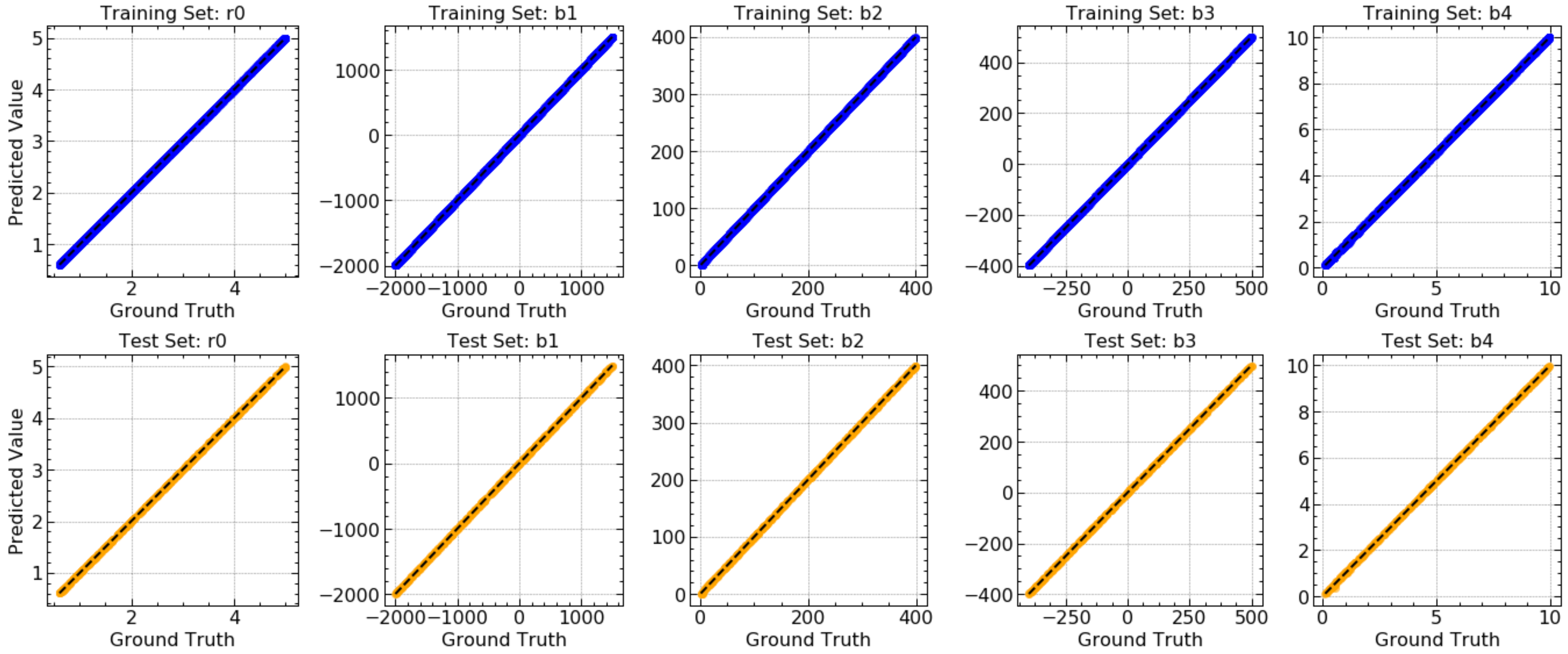


60 points(k), N_C correlations

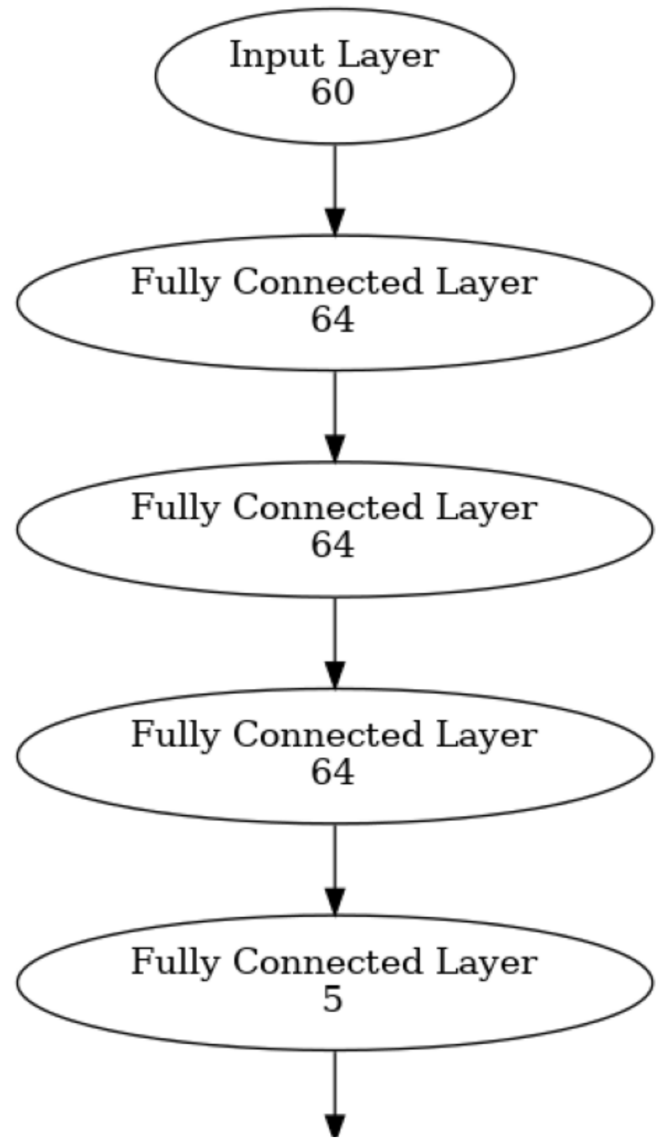
Inverse Femtoscopy

$N_c = 6400$ $r_0 = 1.3\ \textbf{fm}, b_1 = -306.5, b_2 = 200, b_3 = -266, b_4 = 0.78, n_\pi = 2$

in Preparation **with Jiaxing Zhao, etc.**



60 points

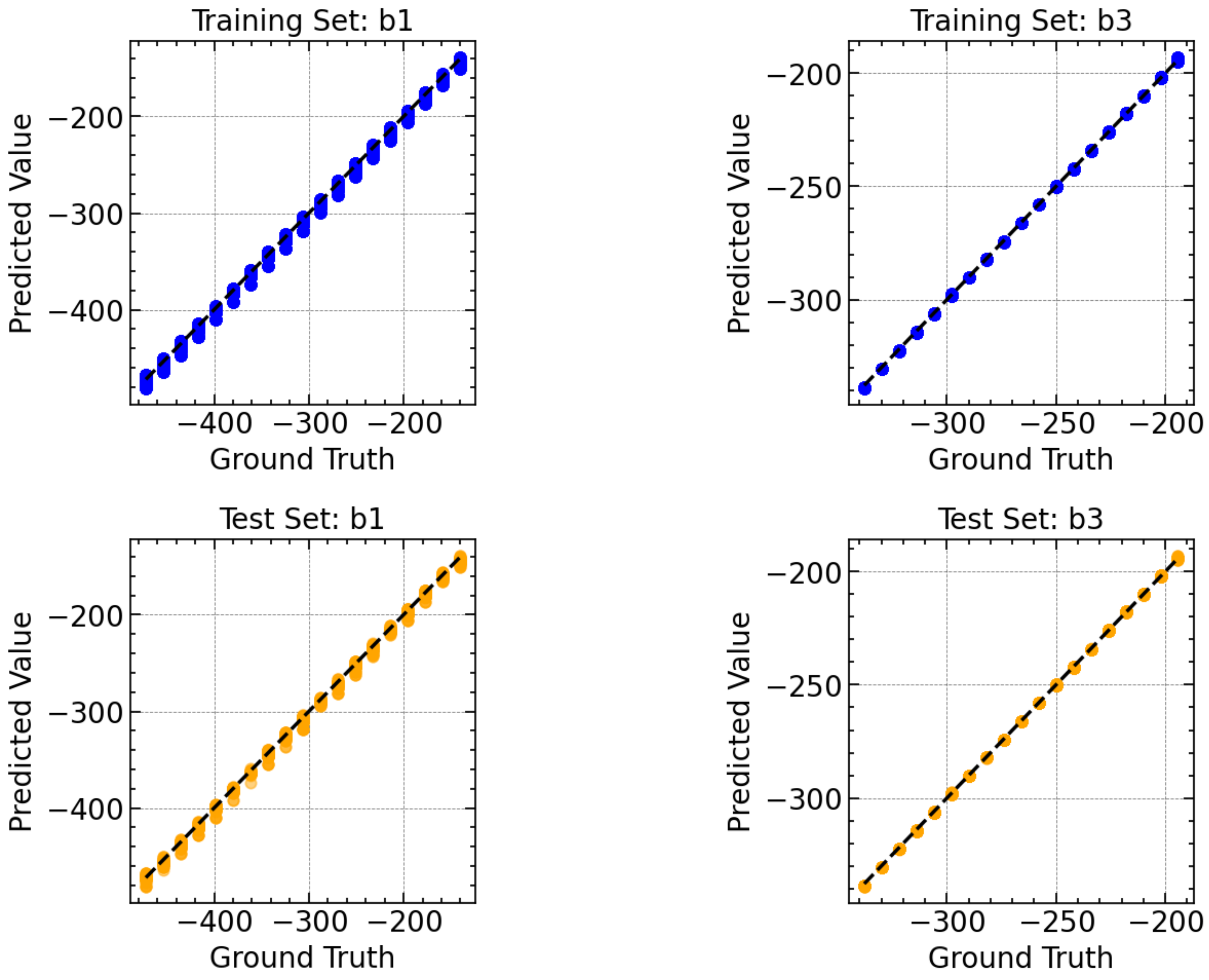


1 parameter

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

R-squared: 0.99, 0.99, 0.99, 0.99, 0.99

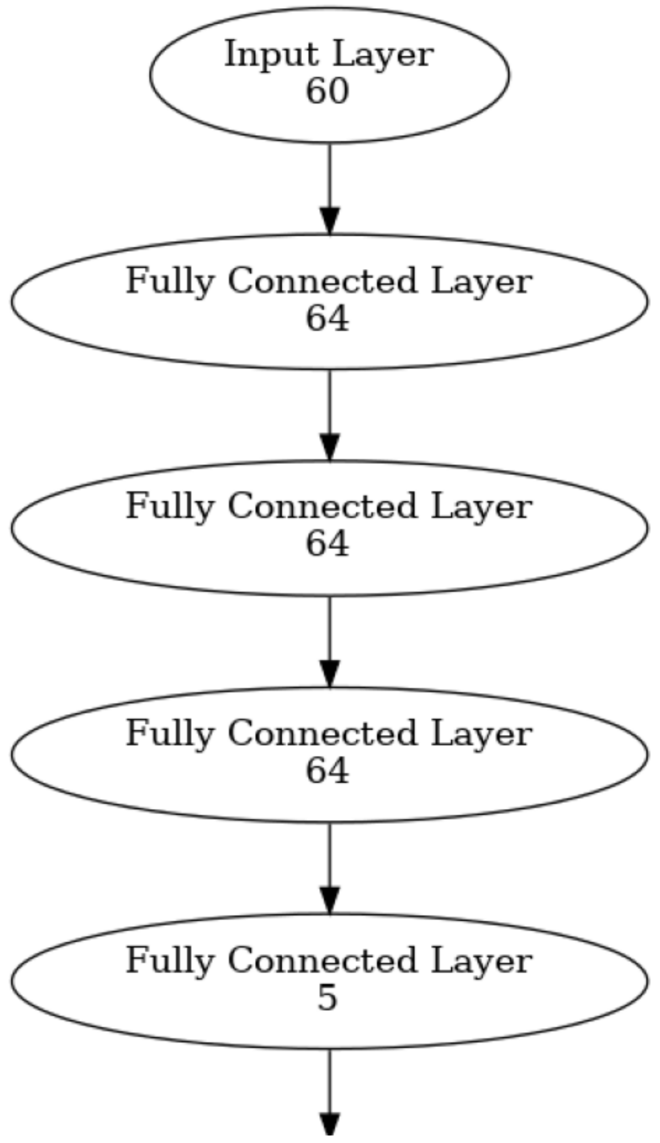
Inverse Femtoscopy



R-squared	b1	b3
Training	0.99	0.99
Testing	0.99	0.99

in Preparation **with Jiaxing Zhao, etc.**

60 points



b_1, b_3

2 parameters

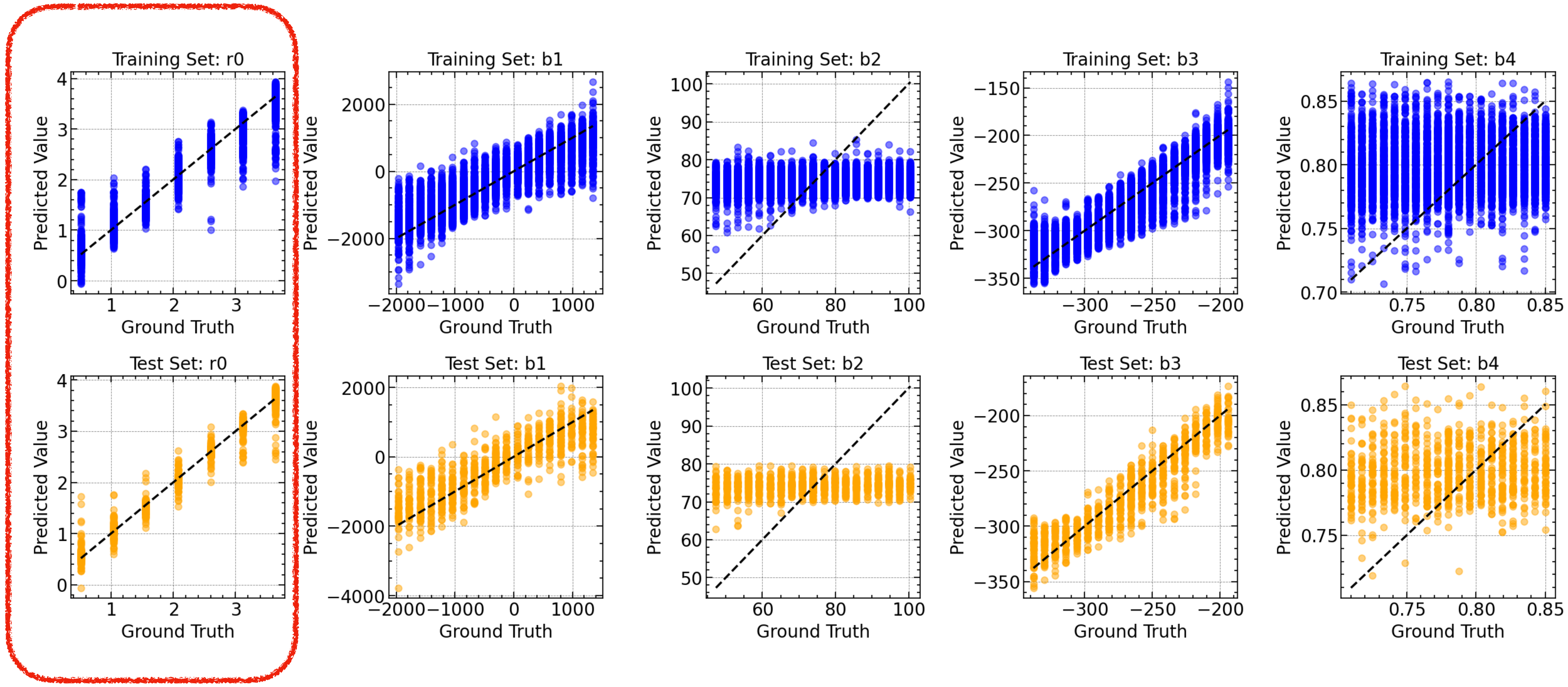
$r_0 = 1.3$ **fm**, $b_2 = 73.9, b_4 = 0.78, n_\pi = 2$

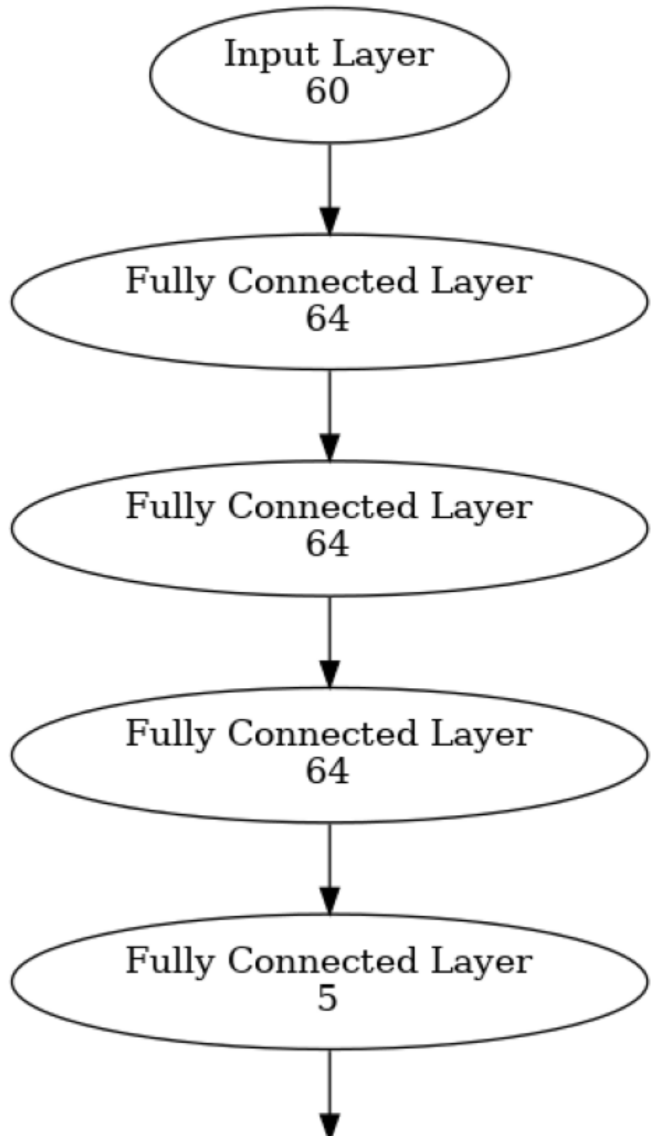
$N_c = 10000$

Inverse Femtoscopy

$N_c = 10000$ $\quad n_\pi = 2$

in Preparation **with Jiaxing Zhao, etc.**





60 points

R-squared	r0	b1	b2	b3	b4
Training	0.99	0.86	0.02	0.90	0.00
Testing	0.99	0.86	0.02	0.90	0.00

5 parameters

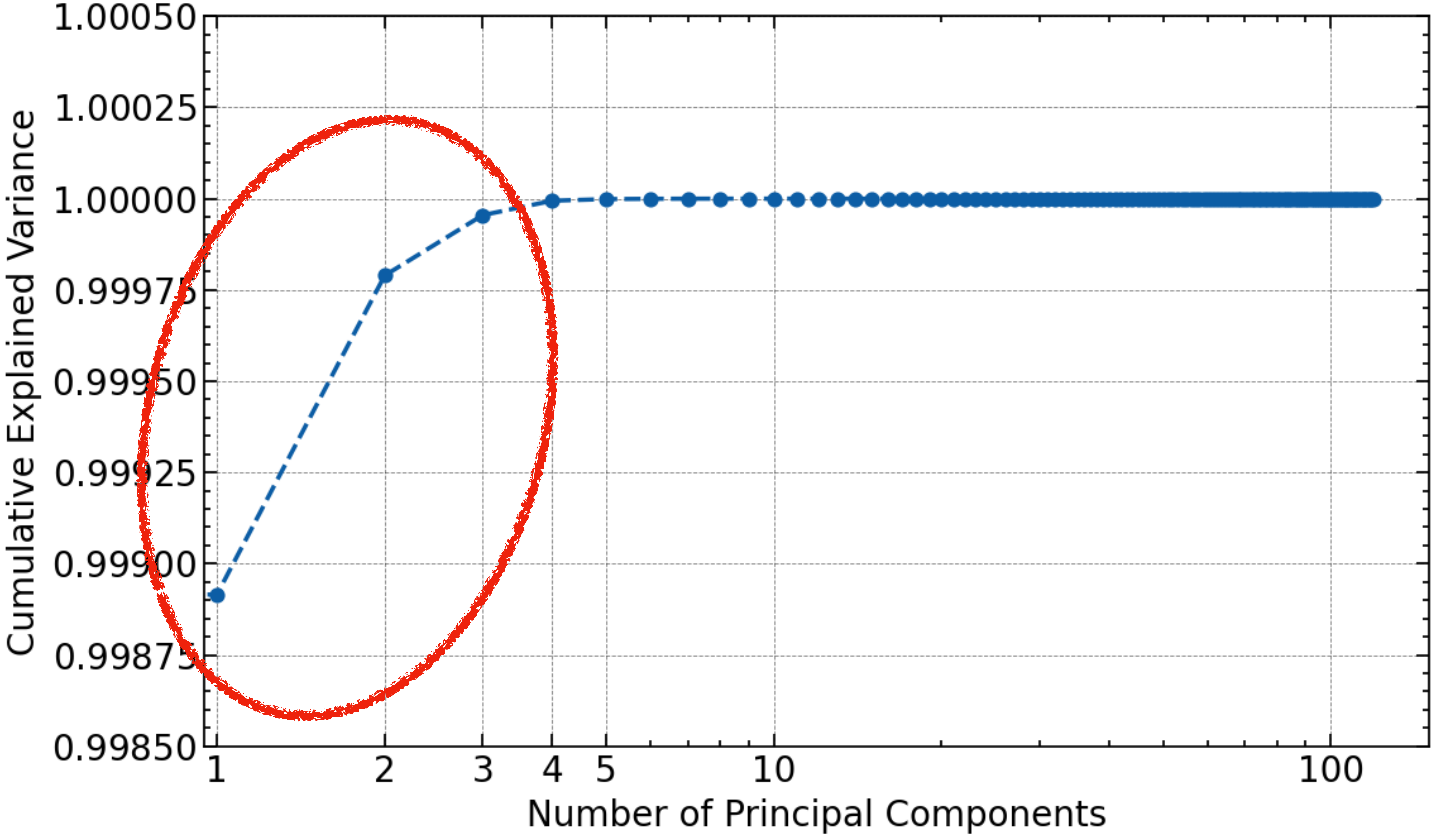
Potential Functions

$$V(r) = b_1 e^{-b_2 r^2} + b_3(1 - e^{-b_4 r^2})(\frac{e^{(-m_\pi r)}}{r})^{n_\pi}$$

Inverse Femtoscopy

in Preparation with Jiaxing Zhao, etc.

Principal Component Analysis(PCA)



$N_c = 25000$

r_0: [0.52,4.16] **fm**

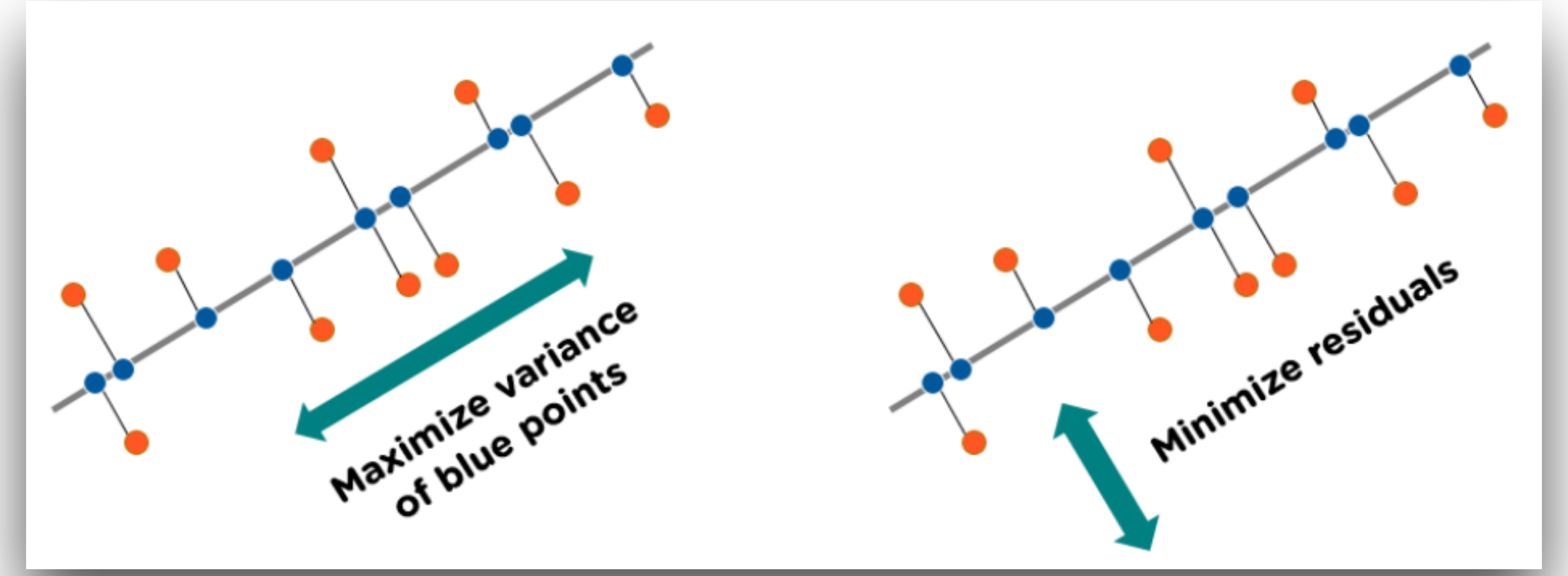
b_1: [−2145.5,1532.5] **MeV**

b_2: [0.739,147.761] **fm**$^{-2}$

b_3: [−1064,532] **MeV** $\cdot$ **fm**^{2}

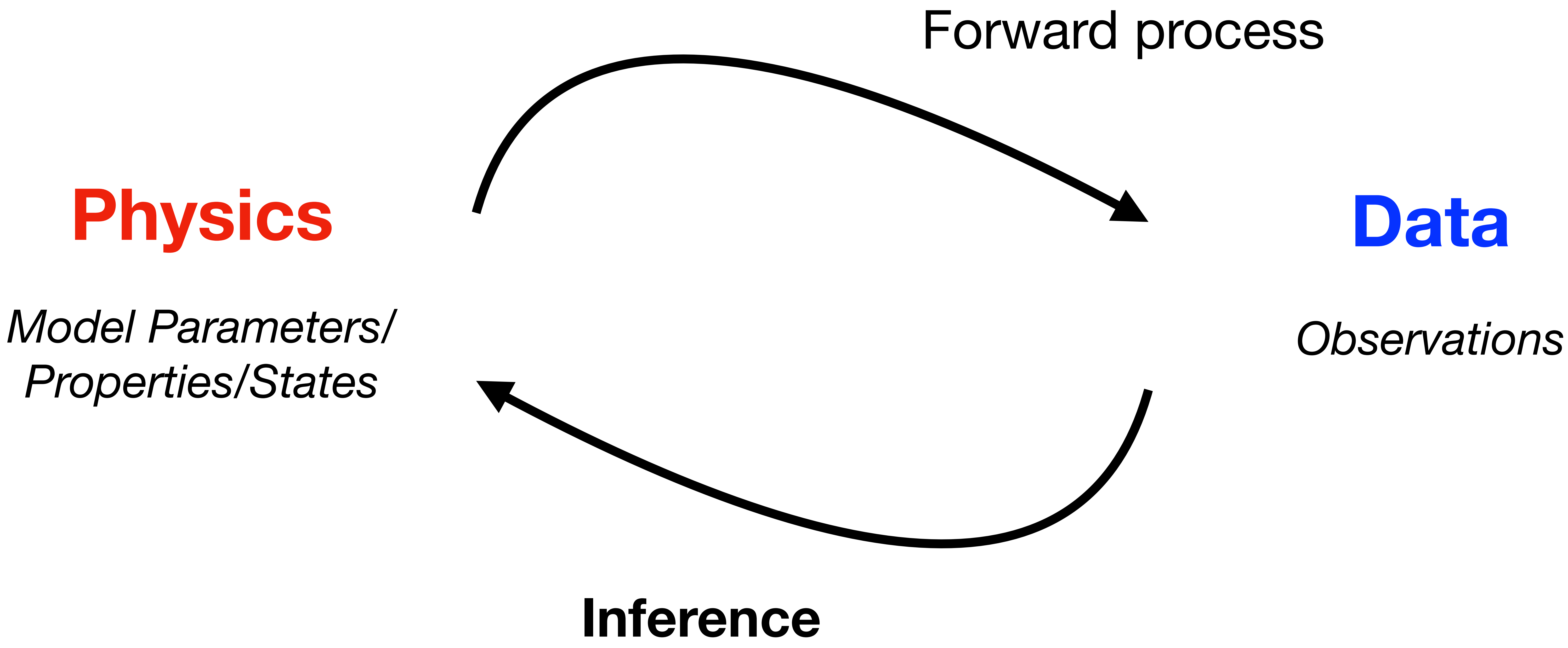
b_4: [0.078,154.422] **fm**$^{-2}$

n_π: 2



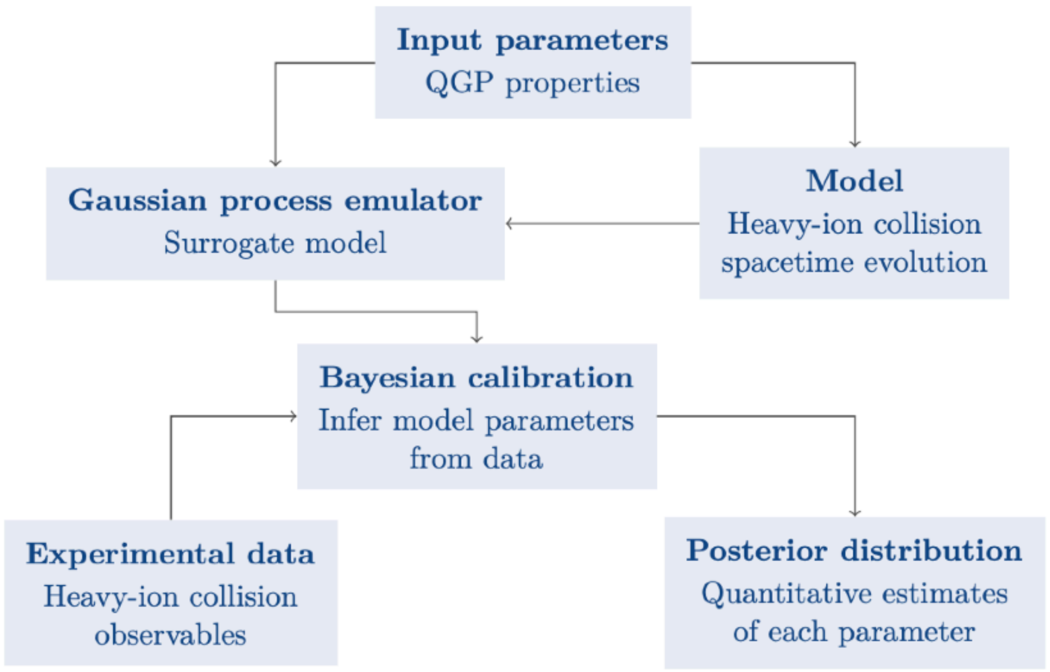
Bayesian Inference

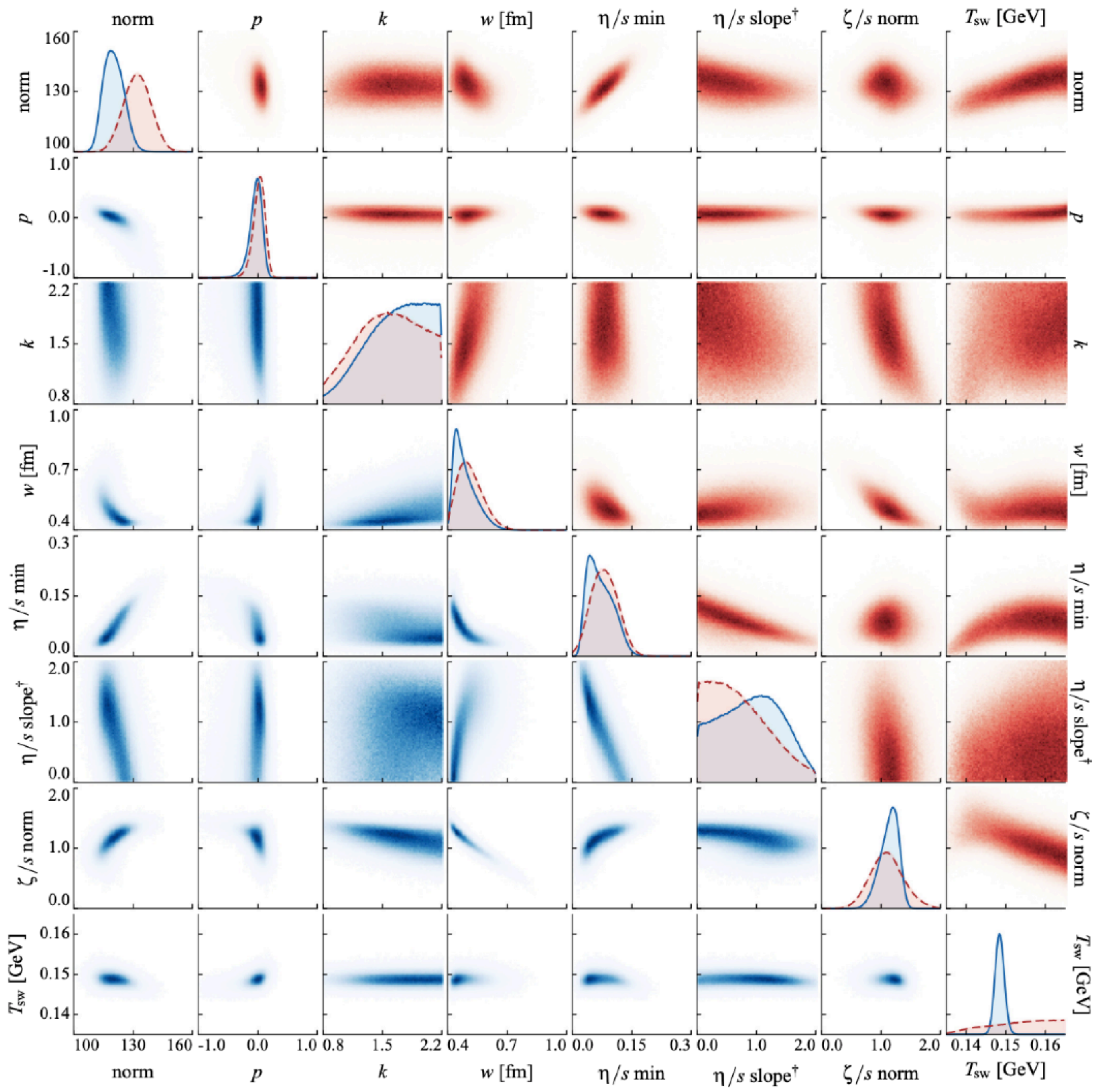
$$\hat{\theta} = \textbf{arg max}_{\theta}\{p(X \mid \theta)\}$$



Bayesian Inference

Inferring Dynamical Information

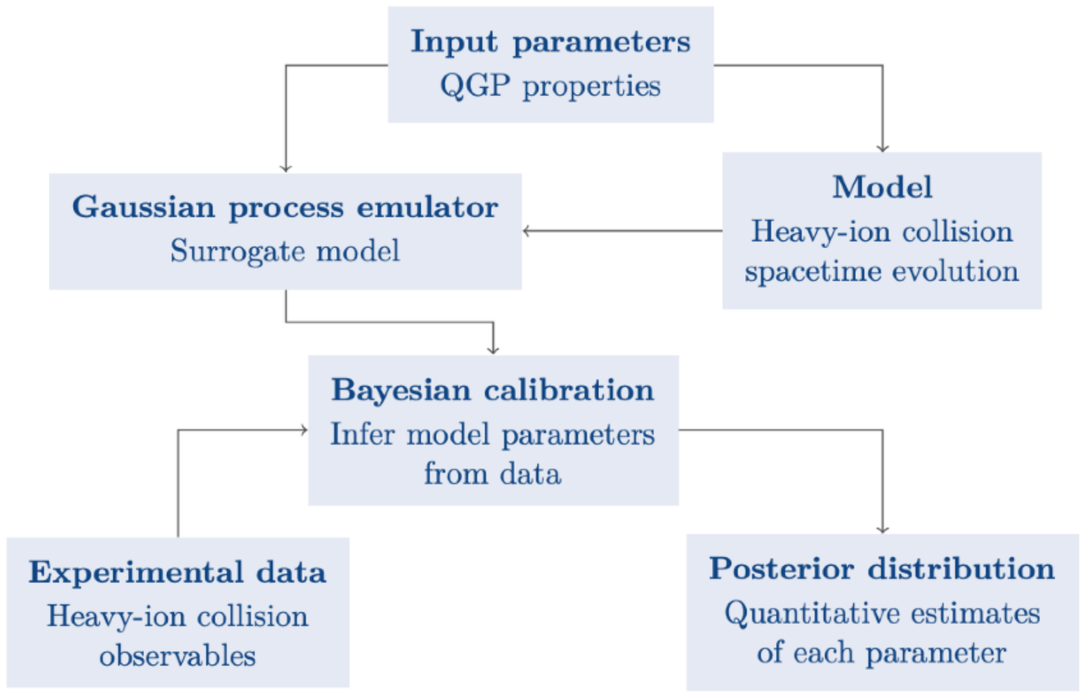




J. E. Bernhard, etc.(Duke Group), arXiv:1804.06469 and Nat Phy **15**, 1113 (2019).

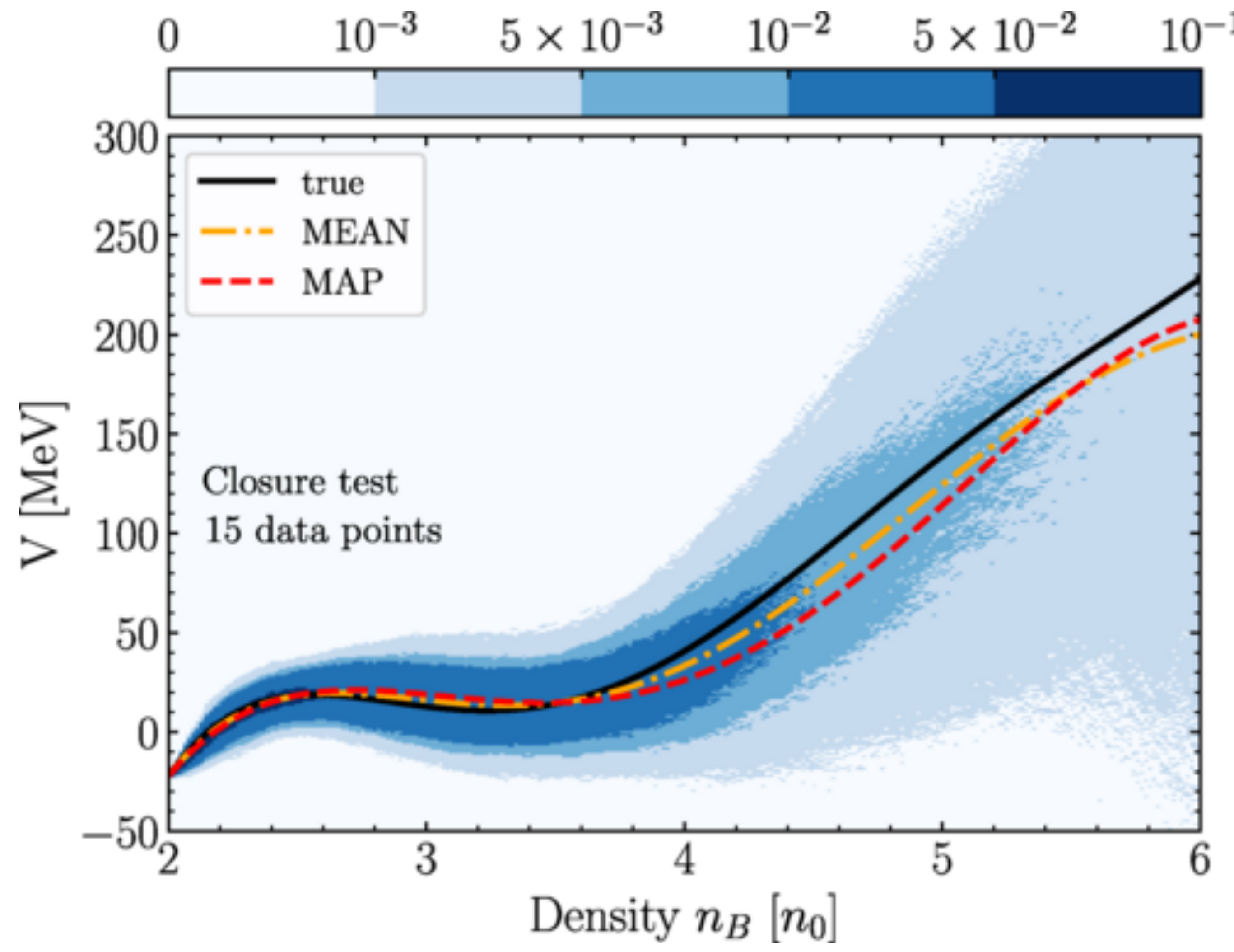
Bayesian Inference

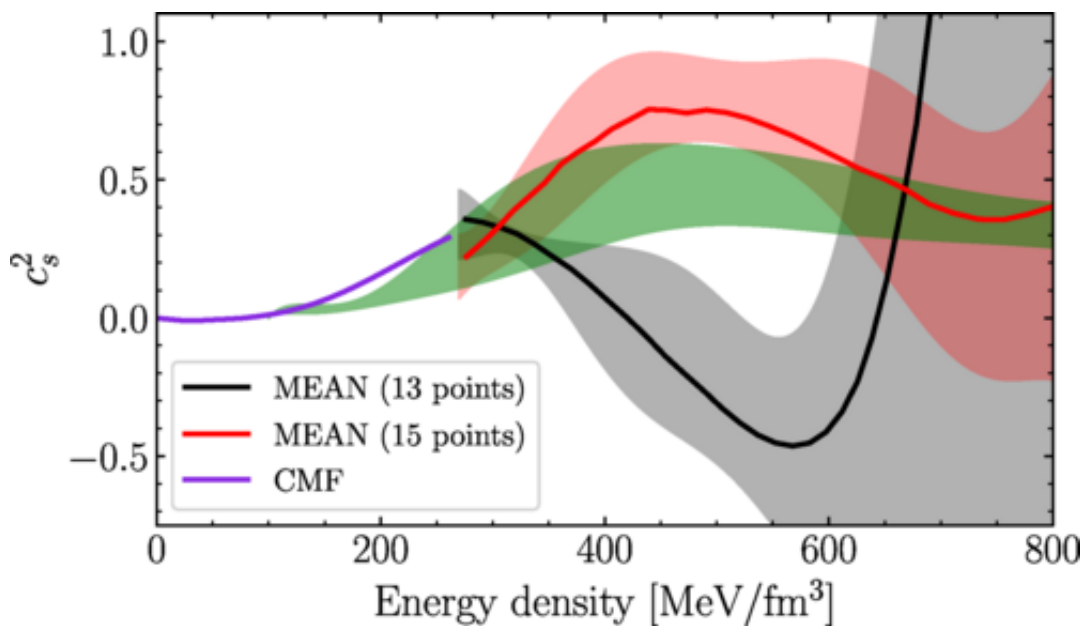
Inferring Dynamical Information



J. E. Bernhard, etc.(Duke Group), arXiv:1804.06469

Ectracting Dense Matter EoSs

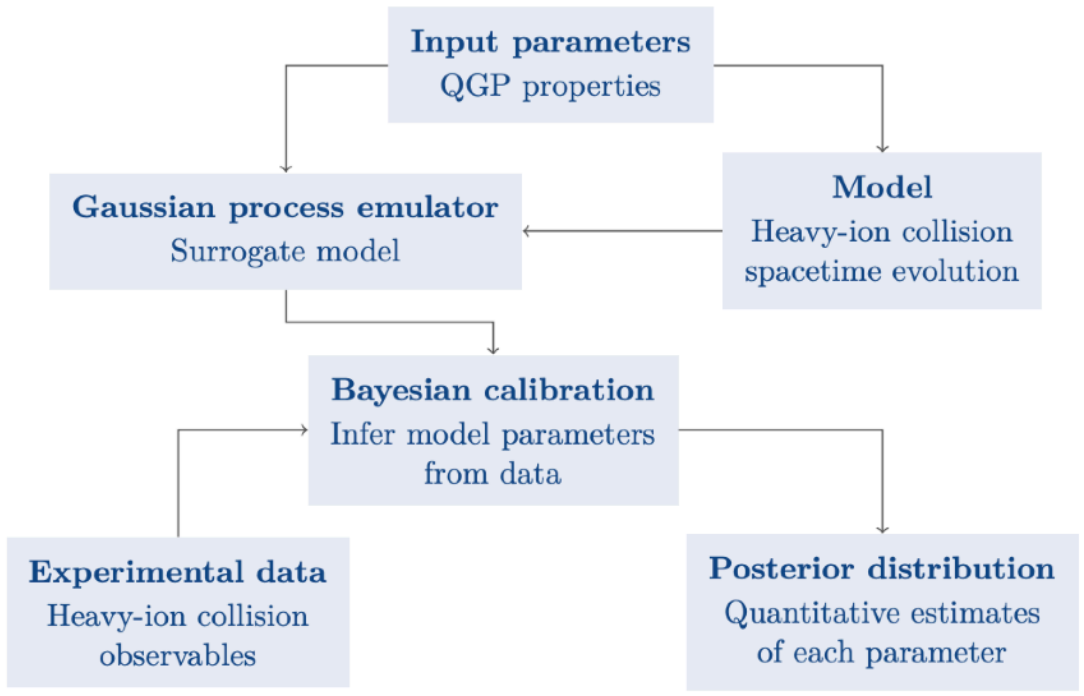




M. Omana Kuttan, J. Steinheimer, K. Zhou, and H. Stoecker, Phys. Rev. Lett. **131**, 202303 (2023).

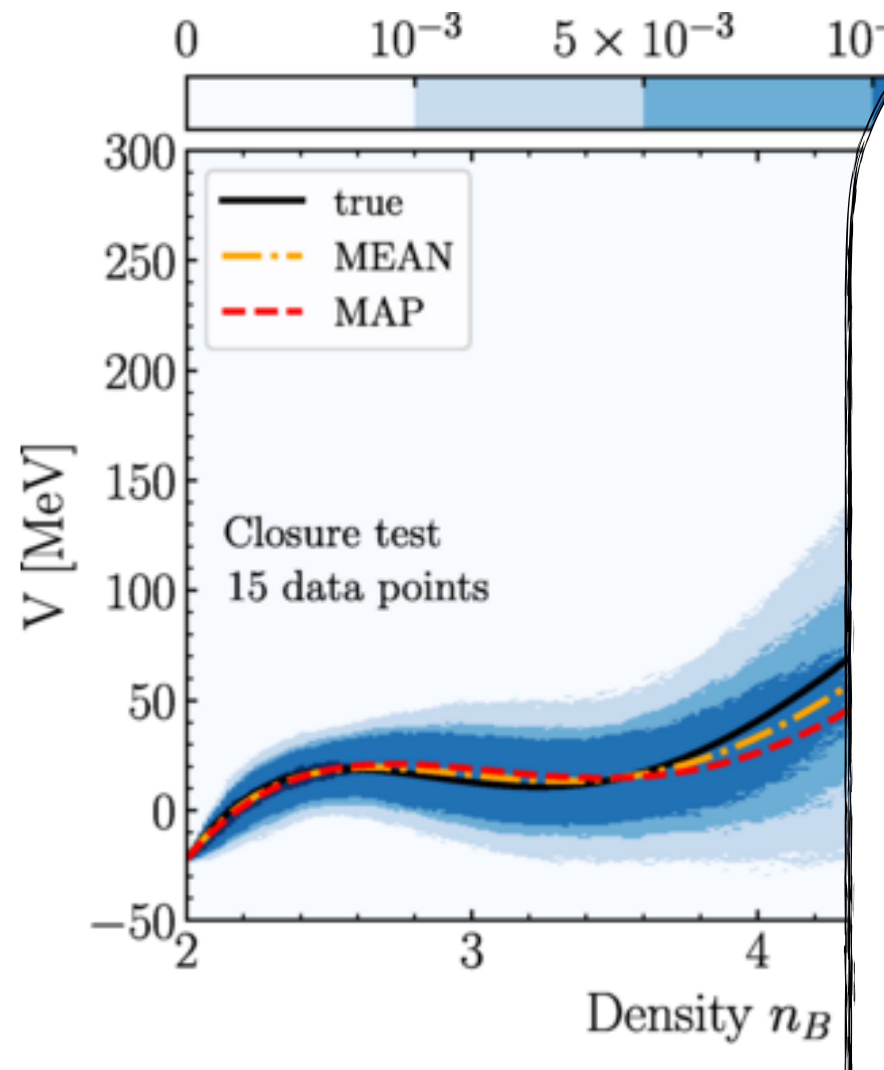
Bayesian Inference

Inferring Dynamical Information



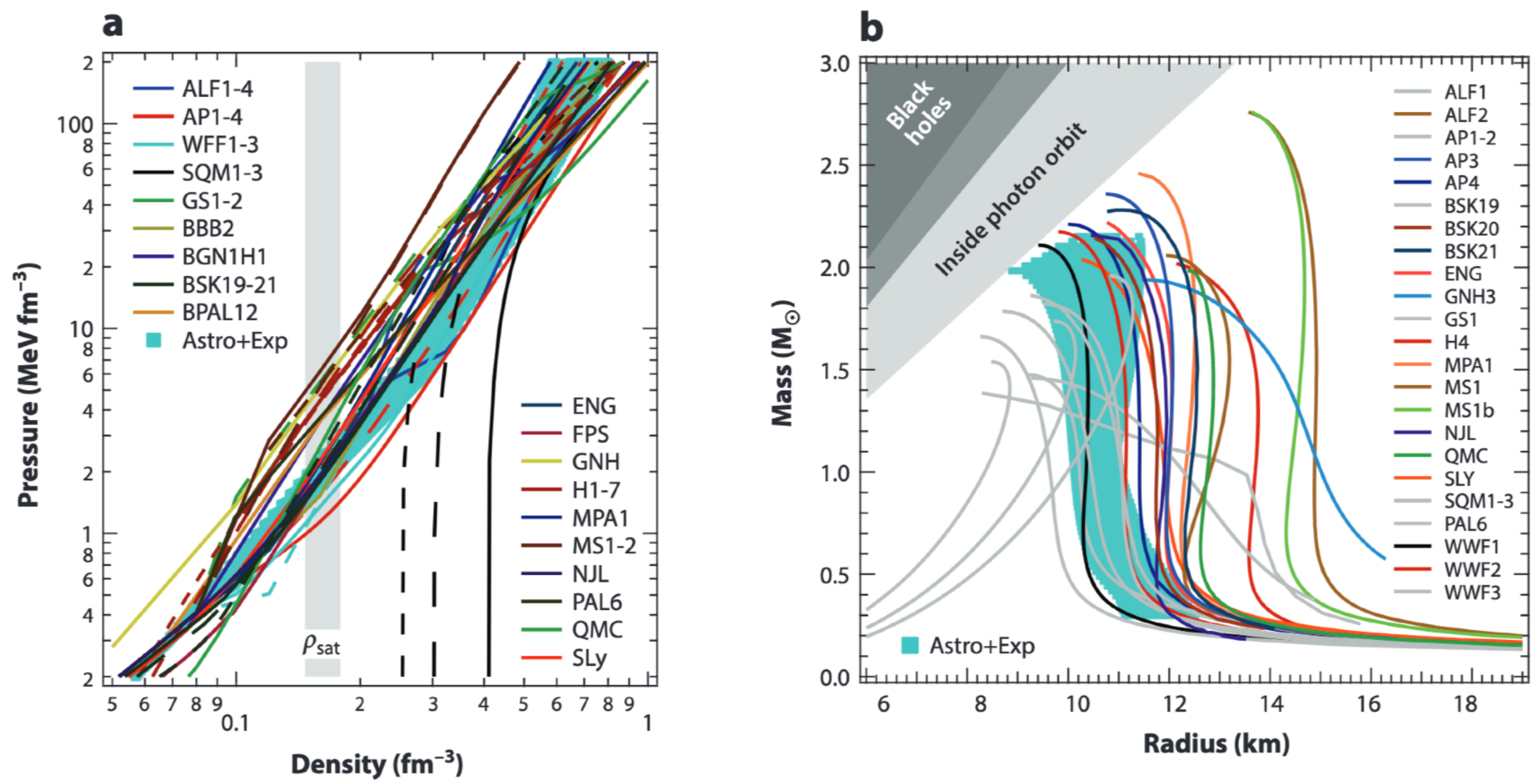
J. E. Bernhard, etc.(Duke Group), arXiv:1804.06469

Ectracting Dense Matter EoSs



M. Omana Kuttan, J. Ste

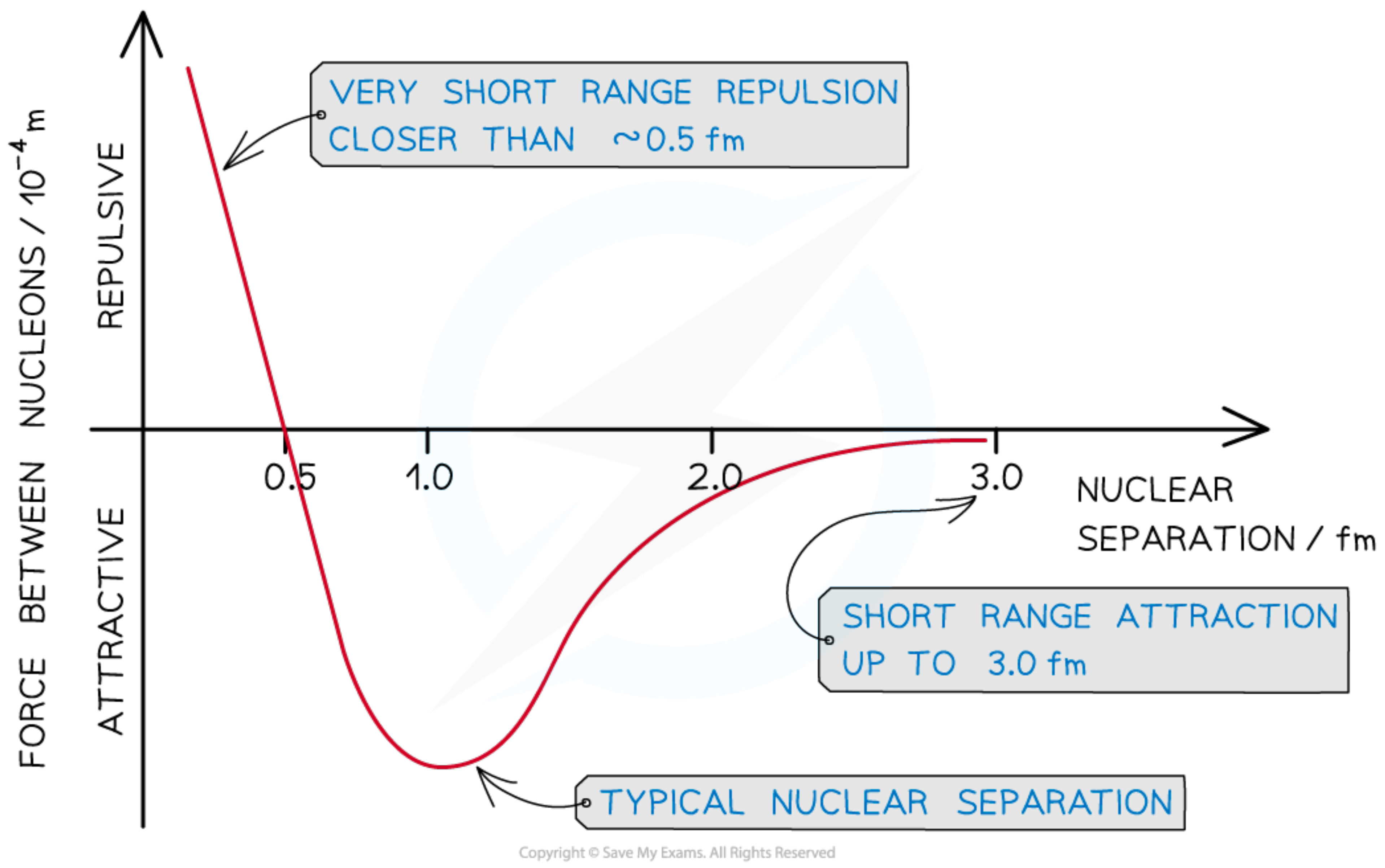
Building Nuclear Matter EoS



F. Özel and P. Freire, Annu. Rev. Astron. Astrophys. **54**, 401 (2016)

Bayesian Inference

Building Nuclear Forces(Interactions)



Physics Parameters are Finite

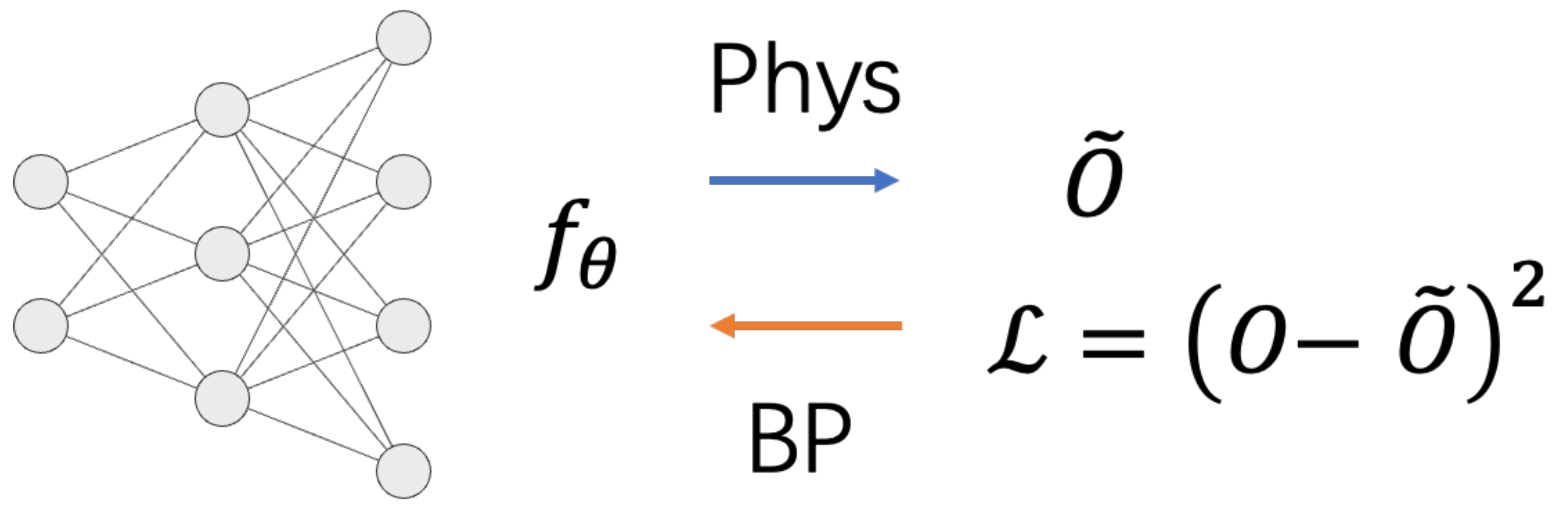
EoS, Wave-Function, **Potential**, ✗

Inference is Easy-To-Compute

ODEs, PDEs, Simulations, … ✗

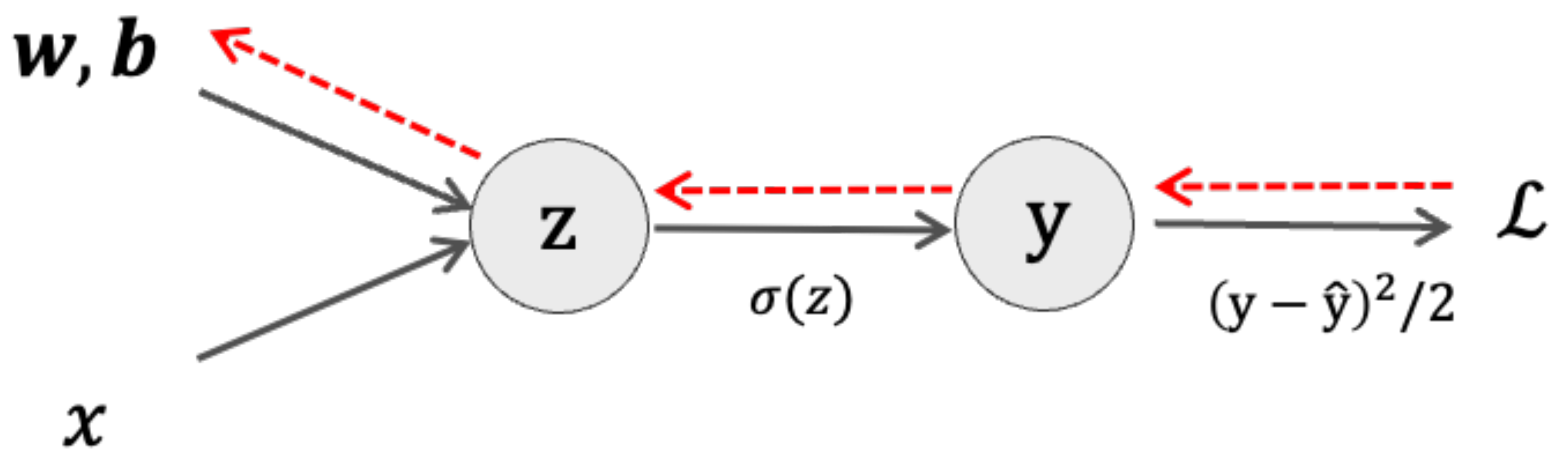
Physics-Driven Deep Learning

$$\hat{\theta} = \textbf{arg max}_{\theta}\{p(X \mid \theta)\}$$



Deep Neural Network represented Physics, f_θ

Flexible Representation

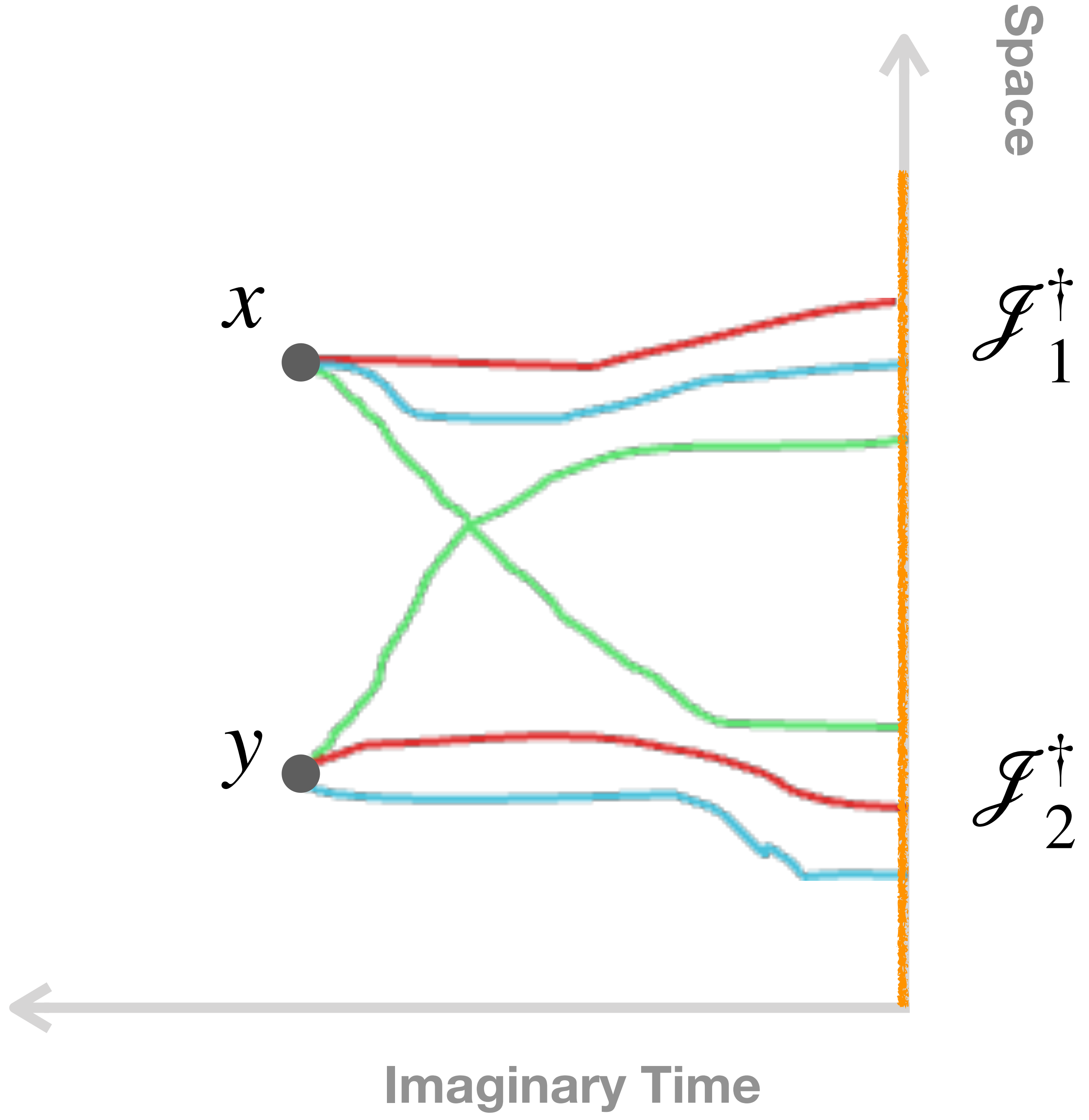


Back-Propagation

Easy-To-Compute on GPUs

HAL QCD method

N. Ishii, S. Aoki, and T. Hatsuda, Phys. Rev. Lett. 99, 022001 (2007)
S. Aoki, T. Hatsuda, and N. Ishii, Prog. Theor. Phys. 123, 89 (2010)
Aoki, S., Doi, T., Front. in Phys. 8, 307 (2020)
S. Aoki and T. Doi, in Handbook of Nuclear Physics(2023), pp. 1–31

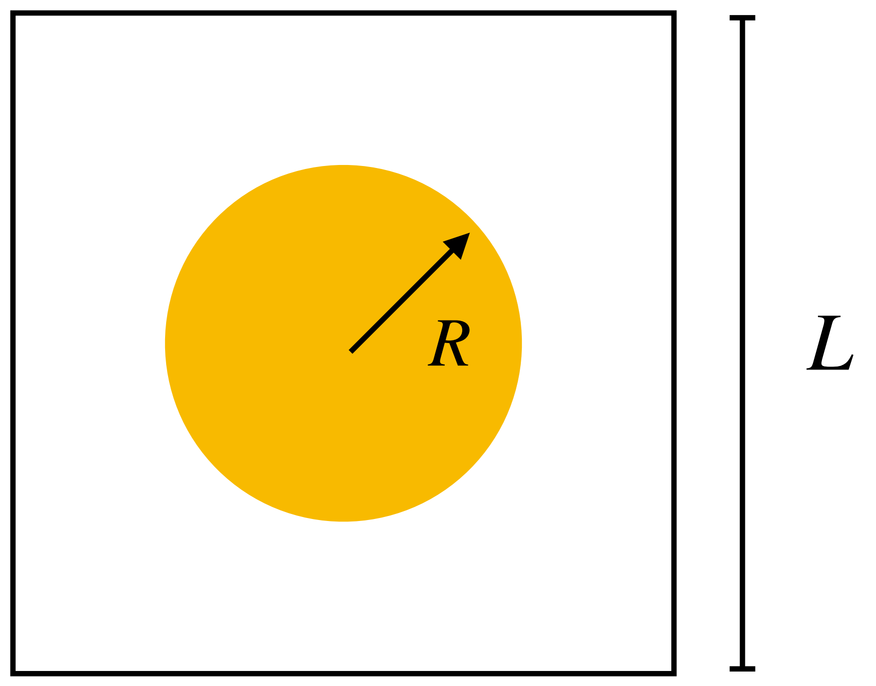


$$\langle N_1(\mathbf{x},t)N_2(\mathbf{y},t)\mathcal{J}_1^\dagger(0)\mathcal{J}_2^\dagger(0)\rangle$$

$$= \sum_n \langle 0|N_1(\mathbf{x})N_2(\mathbf{y})|n\rangle a_n e^{-E_n t}$$

$$\xrightarrow{t>t^*} \phi(\mathbf{r},t) = \sum_{n<n^*} b_n \phi_n(\mathbf{r}) e^{-E_n t}$$

$$(E_k - H_0)\phi_\mathbf{k}(\mathbf{r}) = \int d^3 r' U(\mathbf{r},\mathbf{r}')\phi_\mathbf{k}(\mathbf{r}'), \quad r < R$$

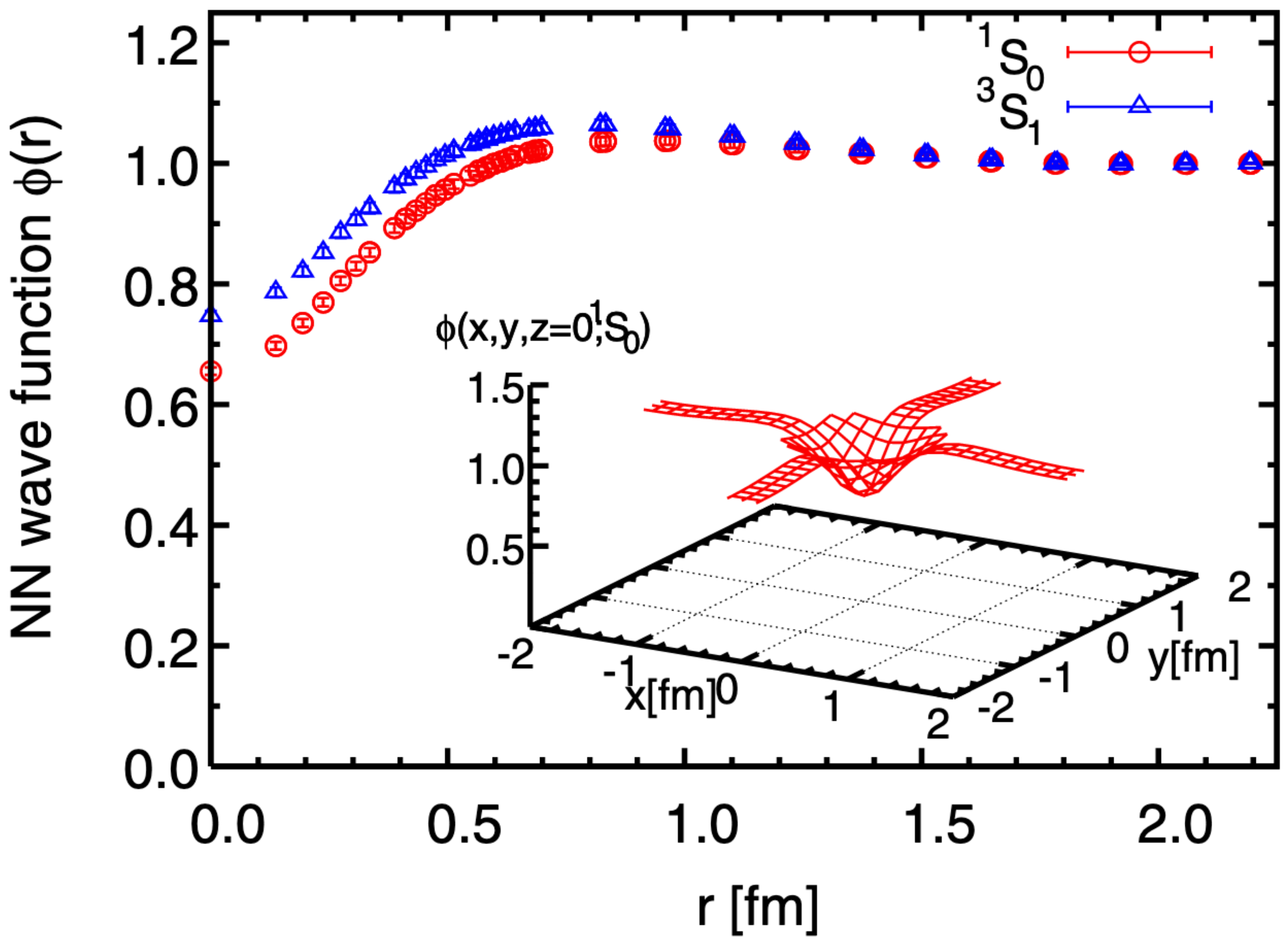


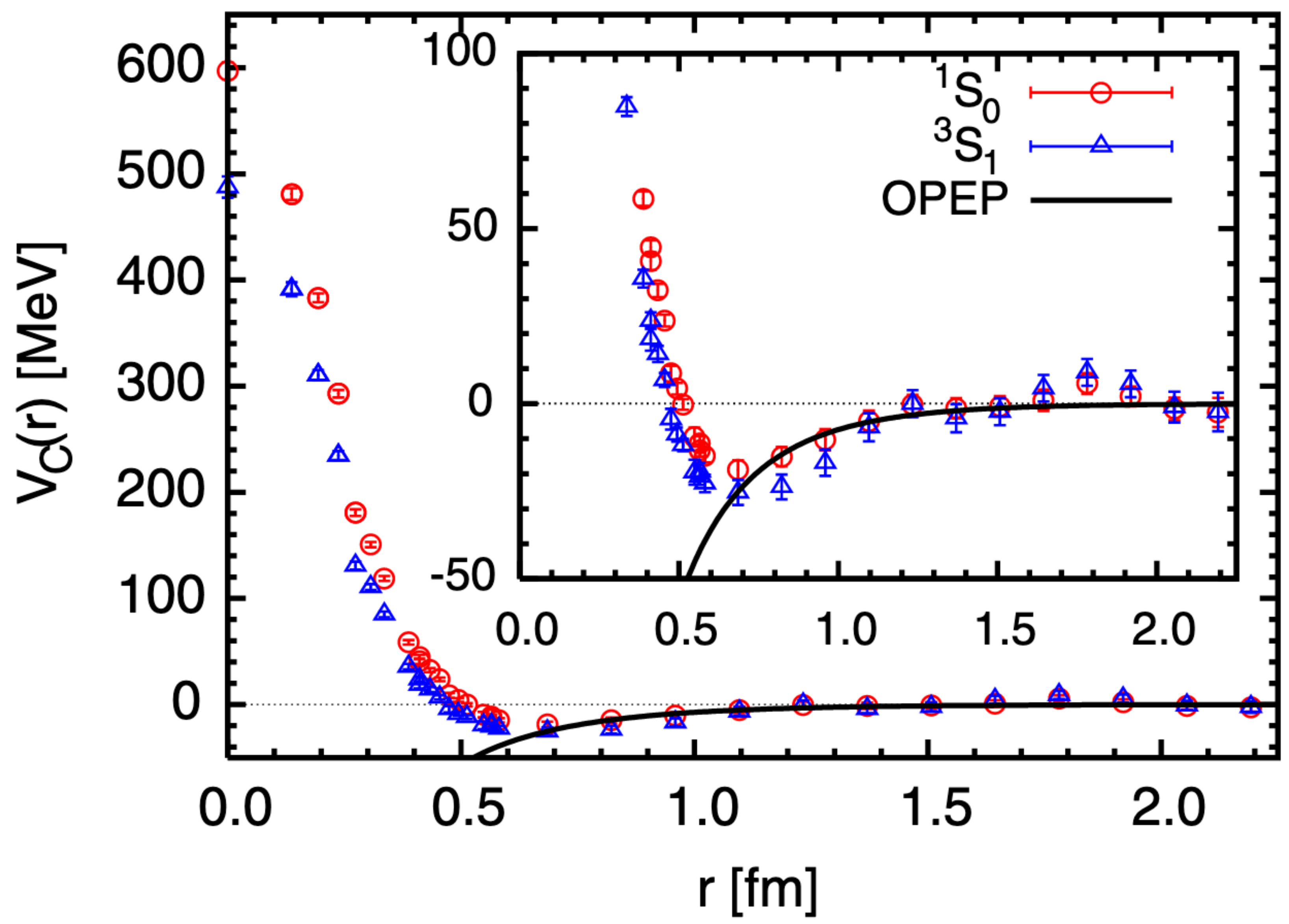
Consider the wave function at “interacting region”⟶Phase shift, Binding energy

$\phi(\mathbf{r},t) \rightarrow$ **2 PI Kernel**

HAL QCD method

N. Ishii, S. Aoki, and T. Hatsuda, Phys. Rev. Lett. 99, 022001 (2007)
S. Aoki, T. Hatsuda, and N. Ishii, Prog. Theor. Phys. 123, 89 (2010)
Aoki, S., Doi, T., Front. in Phys. 8, 307 (2020)
S. Aoki and T. Doi, in Handbook of Nuclear Physics(2023), pp. 1–31





Local Approx.
Derivative Expansion

Nambu-Bethe-Salpeter (NBS) wave function

$$\psi_{NBS}(\vec{r}) = \langle 0|N(\vec{r})N(\vec{0})|N(\vec{k})N(-\vec{k}), in\rangle \simeq e^{i\delta_l(k)} \sin(kr - l\pi/2 + \delta_l(k))/(kr)$$

(at asymptotic region)

HAL QCD method

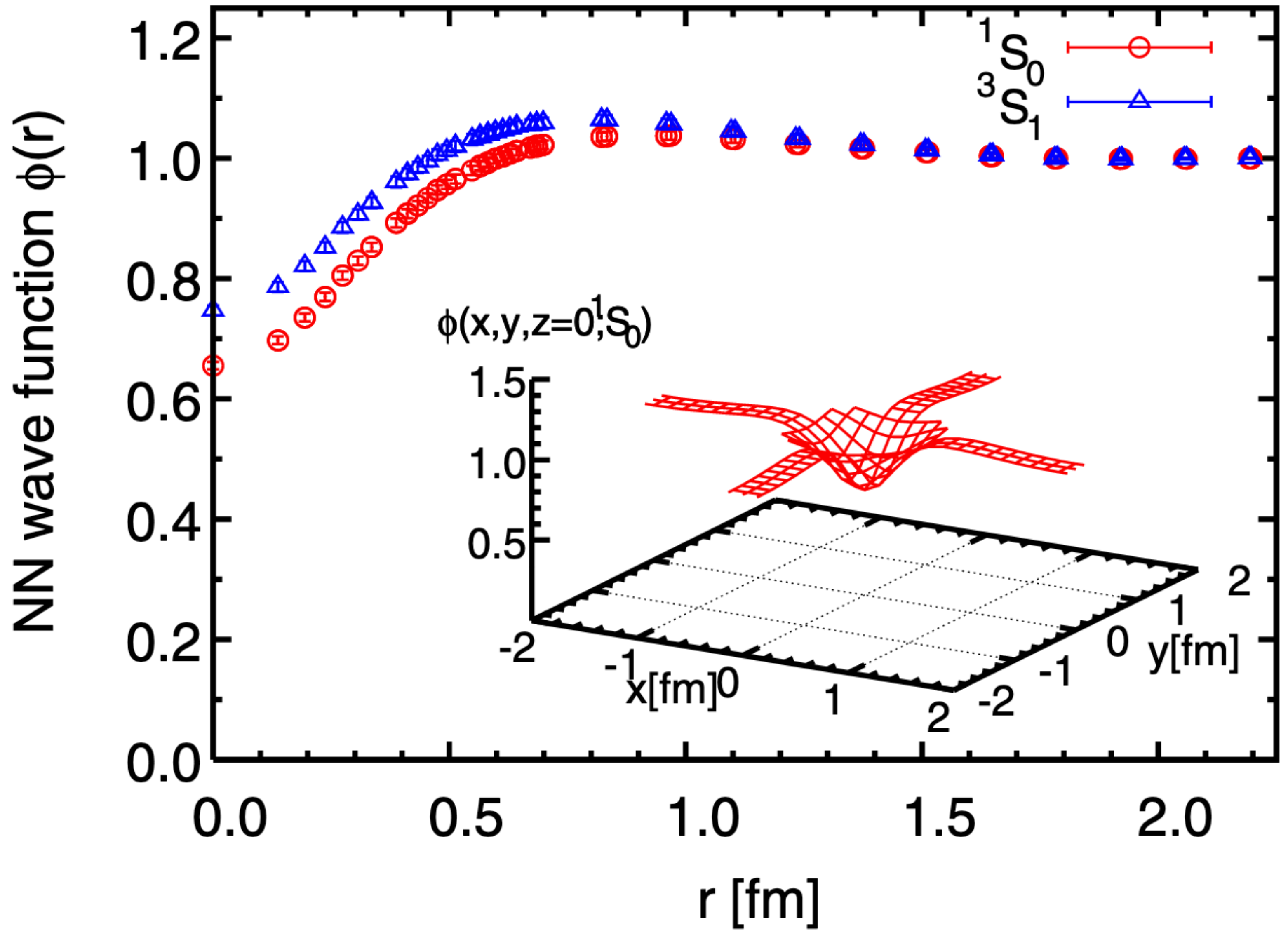
Nulcear Force

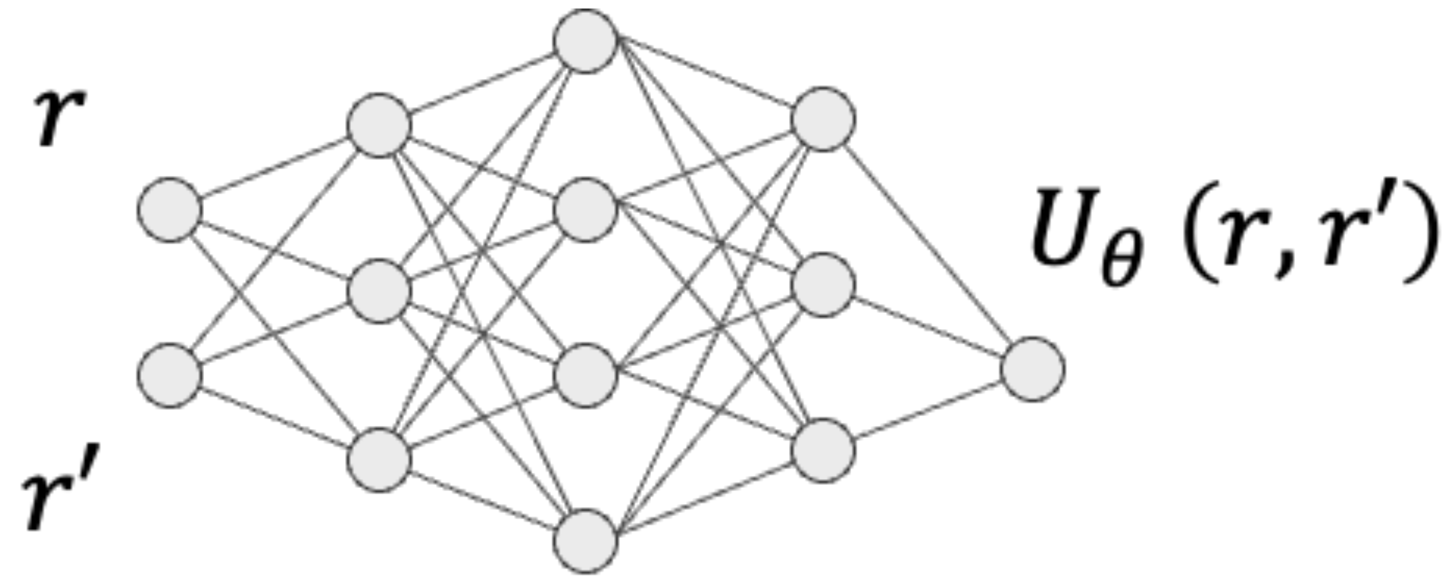
$$(k^2/m_N - H_0)\,\psi_{NBS}(\vec{r}) = \int d\vec{r}' U(\vec{r},\vec{r}')\psi_{NBS}(\vec{r}')$$

(Schrodinger eq.)

HAL QCD: Inverse Problem Perspective

arXiv:2410.03082 (with HAL QCD)





Universal Approximation Theorem (1989,1991)

$$\theta_{i+1} \to \theta_i + \frac{\partial \mathscr{L}}{\partial U_\theta(r,r')}\frac{\partial U_\theta(r,r')}{\partial \theta}$$

Gradient Decent

NBS wave function → **Potential Function**

Data(Observations) ← **Physics Properties**

Maximize Likelihood Estimation

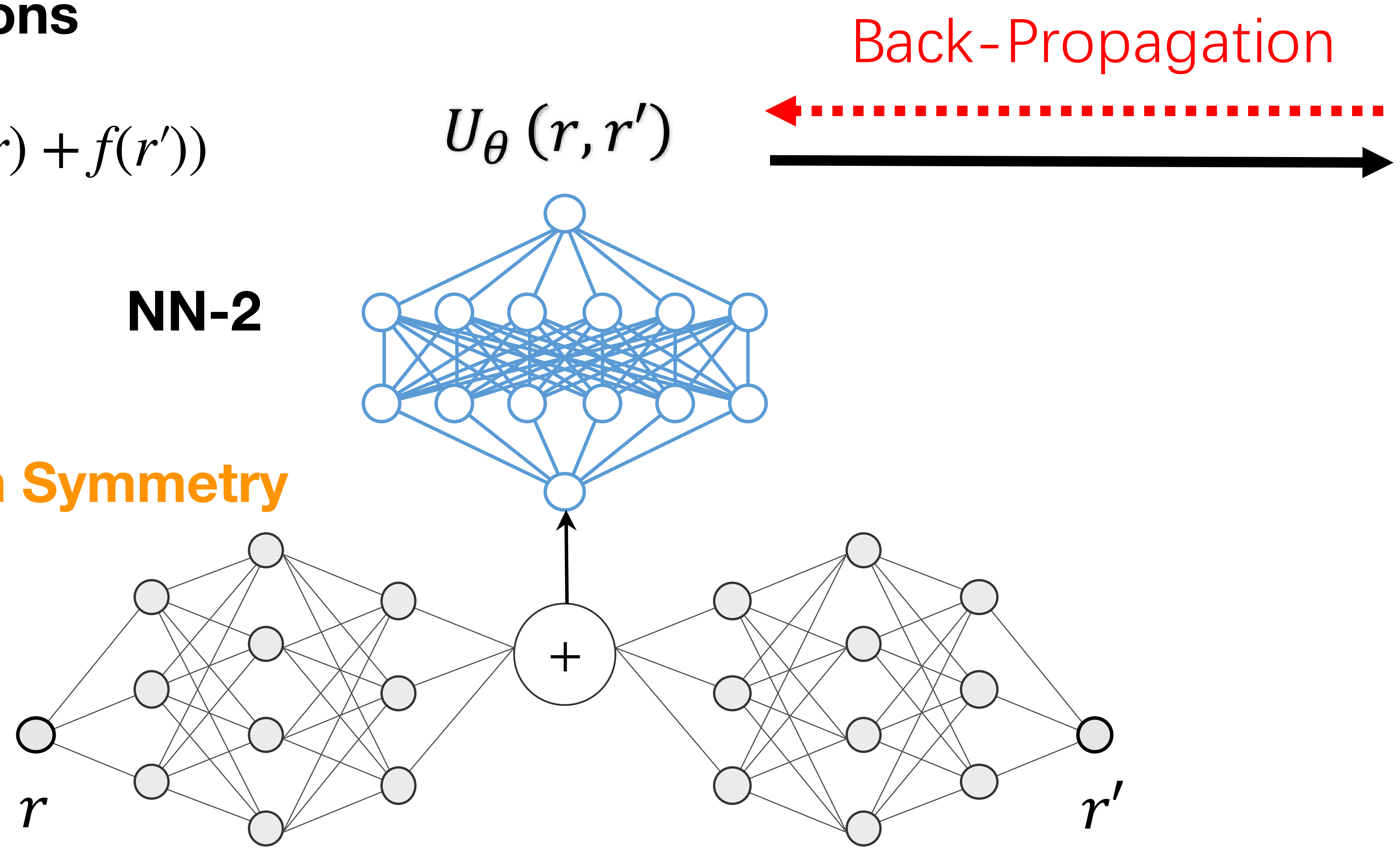
$$\min_\theta \mathscr{L} = \sum_k \int d^3\mathbf{r}\left[\left(E_k - H_0\right)\phi_{\mathbf{k}}(\mathbf{r}) - \int d^3\mathbf{r}' U_\theta(\mathbf{r},\mathbf{r}')\phi_{\mathbf{k}}(\mathbf{r}')\right]^2$$

Physics-Driven Deep Learning

arXiv:2410.03082 (with HAL QCD)

Two particle interactions

$$U_\theta(r, r') \equiv g(f(r) + f(r'))$$



a. Permutation Symmetry

b. Asymptotic Behaviour as regulator

$$\lim_{\mathbf{r}>R, \mathbf{r}'>R} U_\theta(\mathbf{r}, \mathbf{r}') \to 0$$

Residual of Schrödinger Eq.

$$(E_k - H_0)\phi_{\mathbf{k}}(\mathbf{r}) = \int d^3 r' U(\mathbf{r}, \mathbf{r}')\phi_{\mathbf{k}}(\mathbf{r}')$$

$$E_k = \frac{k^2}{2m}, \quad H_0 = -\frac{\nabla^2}{2m}, \quad m = \frac{m_N}{2}$$

$\phi_k\ (r)$

or

$R(t, r)$

Phys. Lett. B 712, 437 (2012)

$$\left\{ \frac{1}{4m_N}\frac{\partial^2}{\partial t^2} - \frac{\partial}{\partial t} - H_0 \right\} R(t, r) = \int 4\pi r'^2 dr'\, U(r, r') R(t, r')$$

Mock Test: Separable Potential

As a numerical example, we take $\mu = 1.0, \omega = -0.017\mu^4, m = 3.30\mu, R = 2.5/\mu$

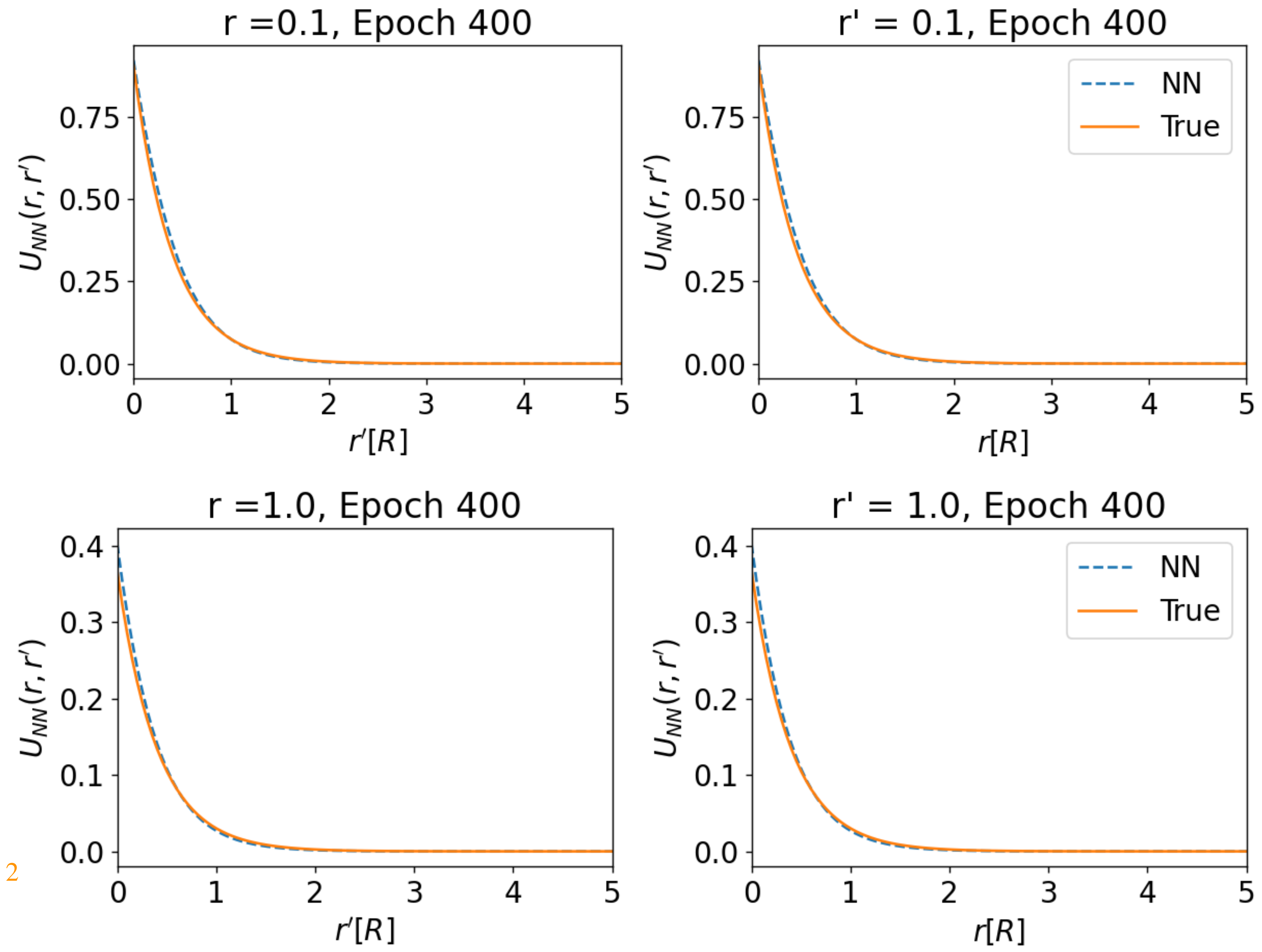
arXiv:2410.03082 (with HAL QCD)

$$U(\mathbf{r}, \mathbf{r}') \equiv \omega\nu(\mathbf{r})\nu(\mathbf{r}'), \quad \nu(\mathbf{r}) \equiv e^{-\mu r}$$

$$U_{\mathbf{NN}}(r, r') = \omega f_\theta(r, r')$$

$$\phi_k^0(r) = \frac{e^{i\delta_0(k)}}{kr}\left[\sin\{kr + \delta_0(k)\} - \sin\delta_0(k) e^{-\mu r}\left(1 + \frac{r(\mu^2+k^2)}{2\mu}\right)\right]$$

$$k\cot\delta_0(k) = -\frac{1}{4\mu^2}\left[2\mu(\mu^2-k^2) - \frac{3\mu^2+k^2}{4\mu^3}(\mu^2+k^2)^2 + \frac{(\mu^2+k^2)^4}{8\pi m\omega}\right]$$



Nambu-Bethe-Salpeter (NBS) wave function

$$\phi_{\mathbf{k}}(\mathbf{r})e^{-W_{\mathbf{k}}t} \equiv \langle 0 | N(\mathbf{x}+\mathbf{r}, t)N(\mathbf{x}, t) | NN, W_k\rangle$$

$$(E_k - H_0)\phi_{\mathbf{k}}(\mathbf{r}) = \int d^3r' U(\mathbf{r}, \mathbf{r}')\phi_{\mathbf{k}}(\mathbf{r}')$$

$$E_k = \frac{k^2}{2m}, \quad H_0 = -\frac{\nabla^2}{2m}, \quad m = \frac{m_N}{2}$$

BP

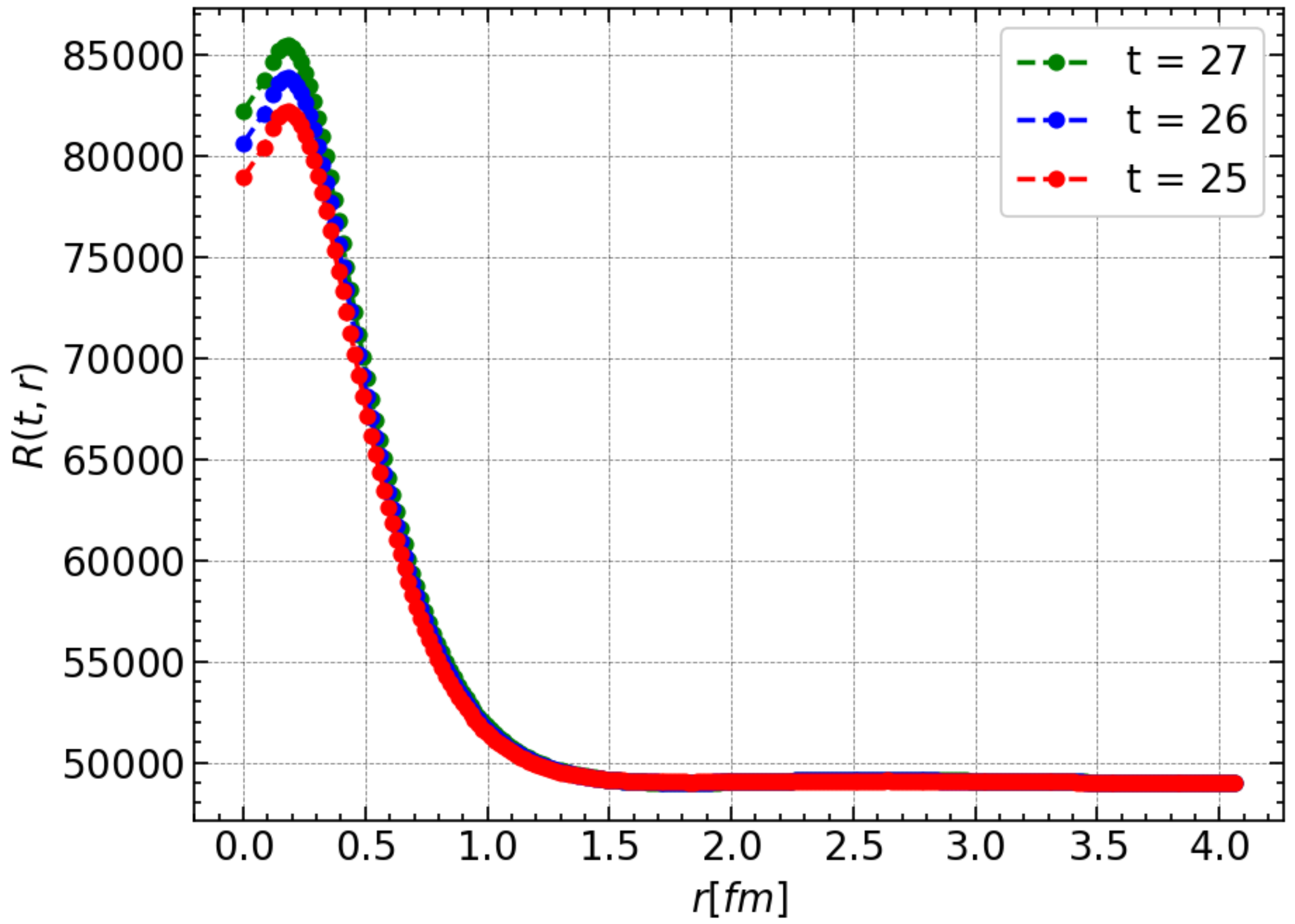
$$\mathcal{L} = \sum_k \int d^3r \left[\left(E_k - H_0\right)\phi_{\mathbf{k}}(\mathbf{r}) - \int d^3r' U_\theta(\mathbf{r}, \mathbf{r}')\phi_{\mathbf{k}}(\mathbf{r}')\right]^2$$

$$U(r > 3R, r' > 3R) \to 0$$

$\Omega_{ccc}\Omega_{ccc}$ Interaction: $^1S_0$

Y. Lyu, etc., Phys. Rev. Lett. 127, 072003 (2021)

$m_N = 2.073, a^{-1} = 2333.0$ **MeV**



$$\left\{ \frac{1}{4m_N} \frac{\partial^2}{\partial t^2} - \frac{\partial}{\partial t} - H_0 \right\} R(t,r) = \int 4\pi r'^2 dr'\, U(r,r') R(t,r')$$

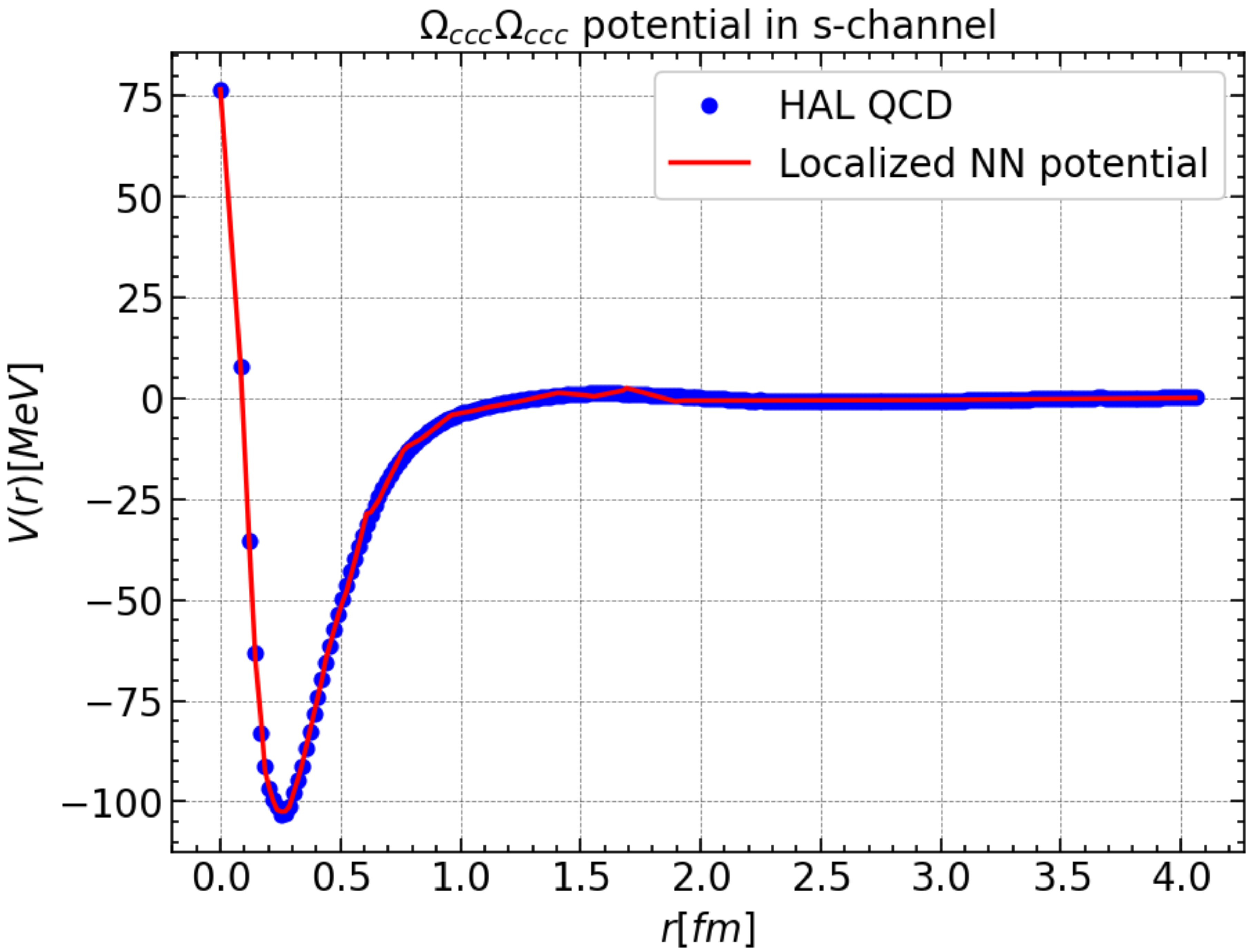
$$R2 = R_{t+1} - 2R_t + R_{t-1}, R1 = (R_{t+1} - R_{t-1})/2, Rr = \nabla^2 R(t,r)$$

BP

Nambu-Bethe-Salpeter (NBS) wave function →

arXiv:2410.03082 (with HAL QCD)

Neural Network Hadron Force



$$\mathcal{L} = \sum_t \left\{ \frac{1}{4m_N} R2(t,r) - R1(t,r) + \frac{1}{m_N} Rr(t,r) - \int 4\pi r'^2 dr'\, U_\theta(r,r') R(t,r') \right\}$$

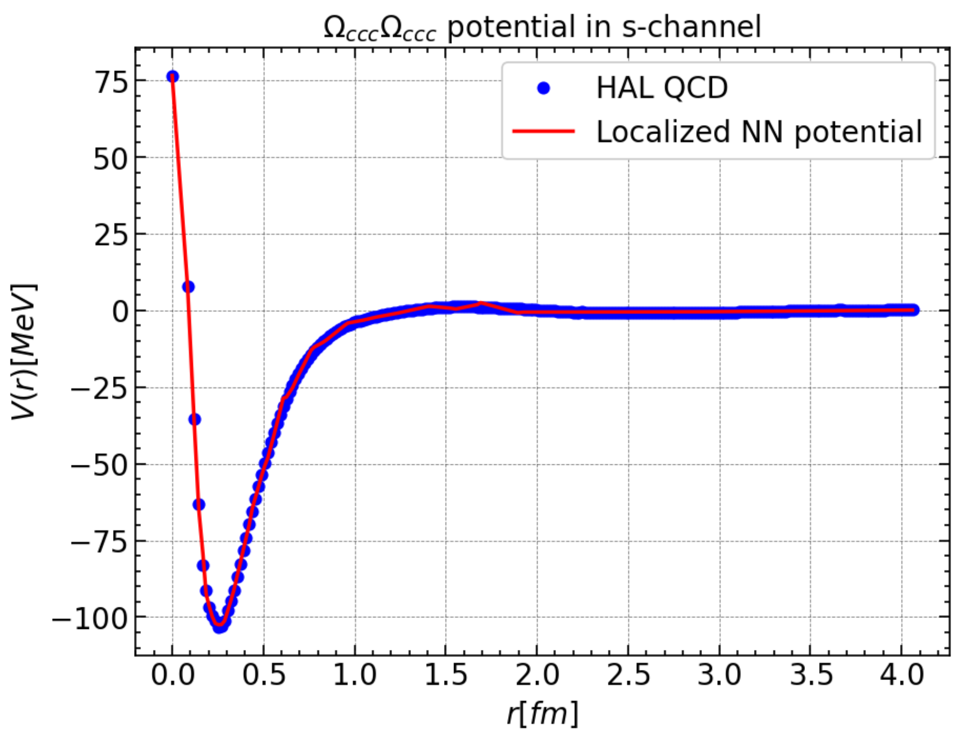
$$U(r > 3\text{fm}, r' > 3\text{fm}) \to 0$$

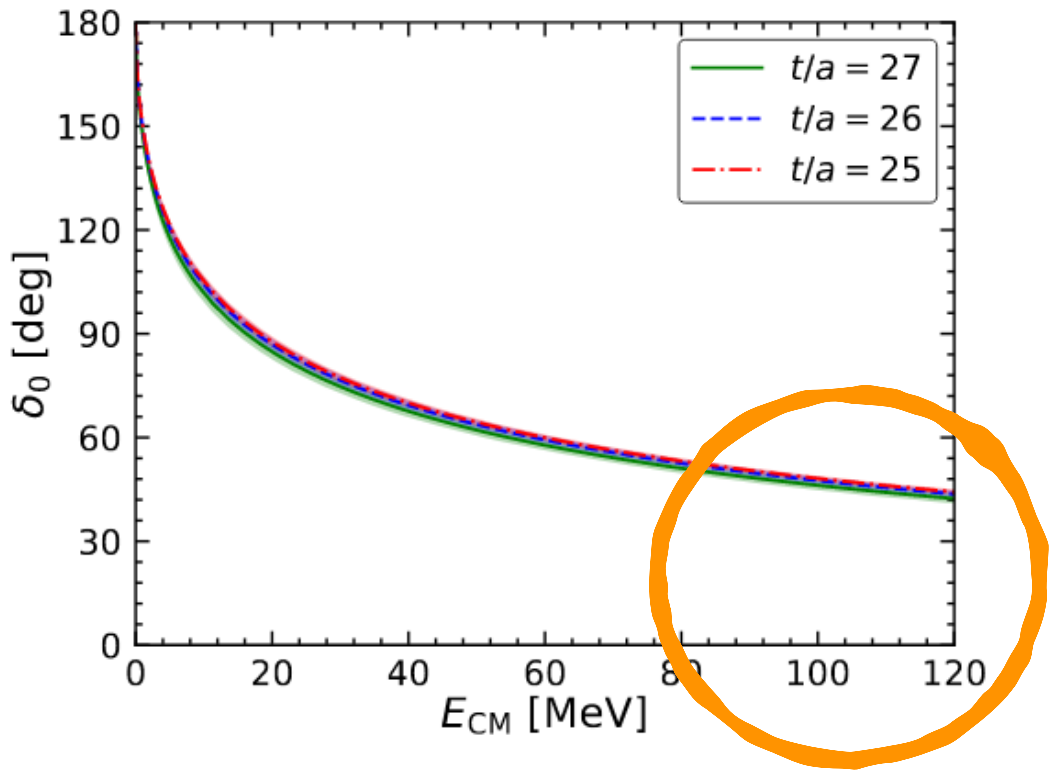
$$V_\theta(r) \equiv \frac{\sum_{r'} \Delta r' 4\pi r'^2 U_\theta(r,r') R(t,r')}{R(t,r)}$$

$\Omega_{ccc}\Omega_{ccc}$ Interaction: Non-Local Potential

arXiv:2410.03082 (with HAL QCD)

Y. Lyu, etc., Phys. Rev. Lett. 127, 072003 (2021)





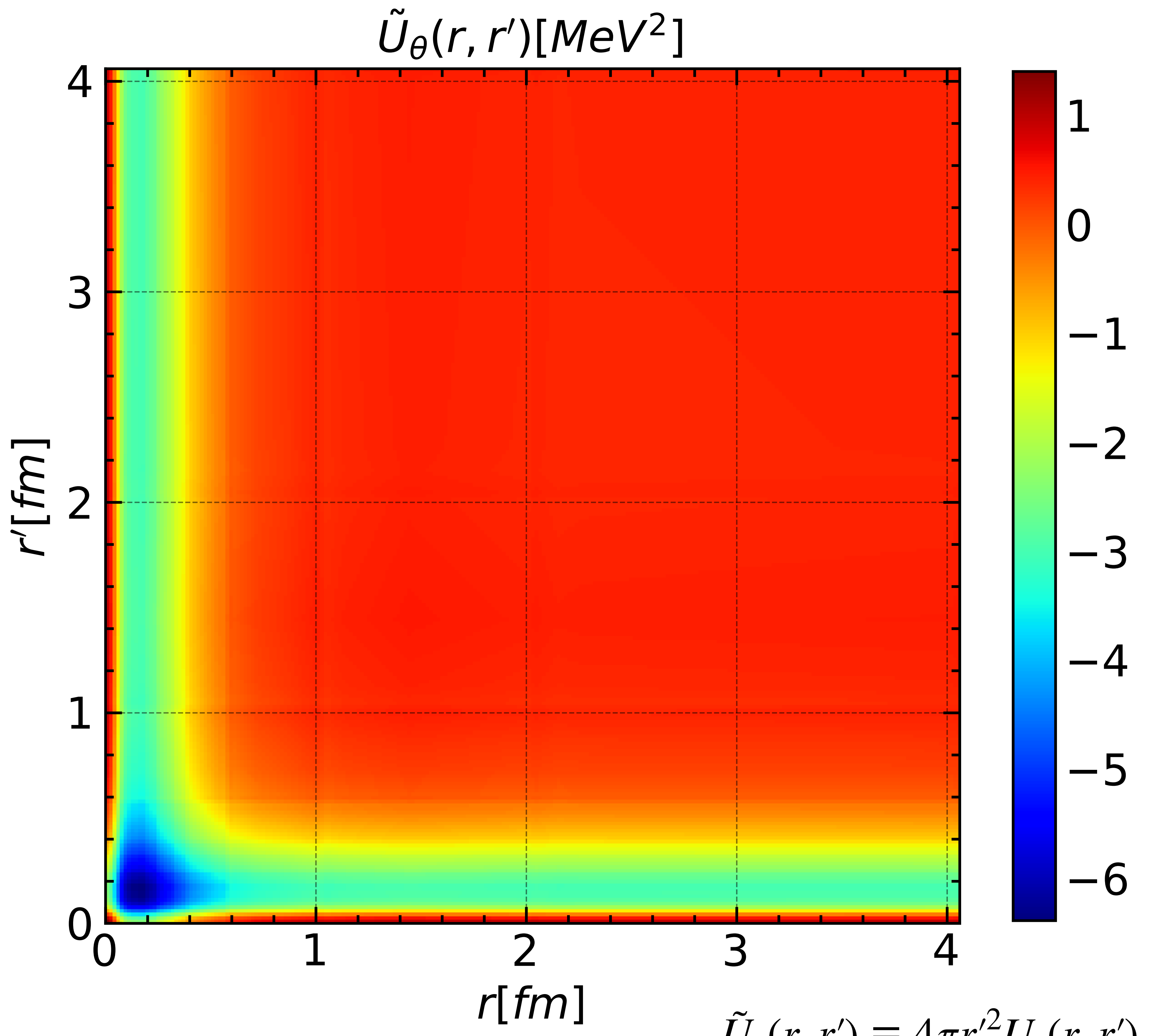
"3D Map"

$$(E_k - H_0)\phi_{\mathbf{k}}(\mathbf{r}) = \int d^3 r' U_\theta(\mathbf{r}, \mathbf{r}')\phi_{\mathbf{k}}(\mathbf{r}')$$

$$E_k = \frac{k^2}{2m}, \quad H_0 = -\frac{\nabla^2}{2m}, \quad m = \frac{m_N}{2}$$

Affect ?

Neural Network Hadron Force

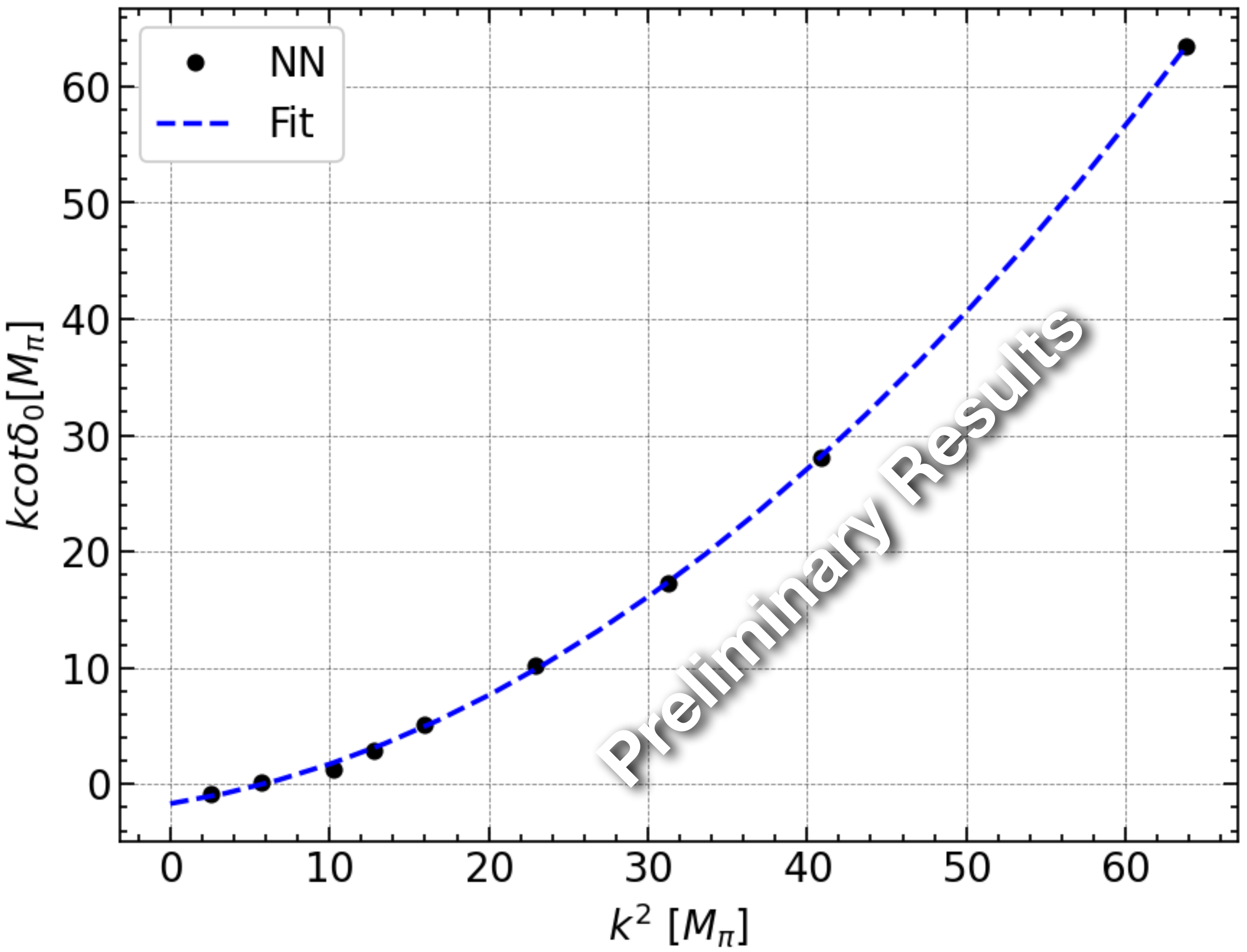


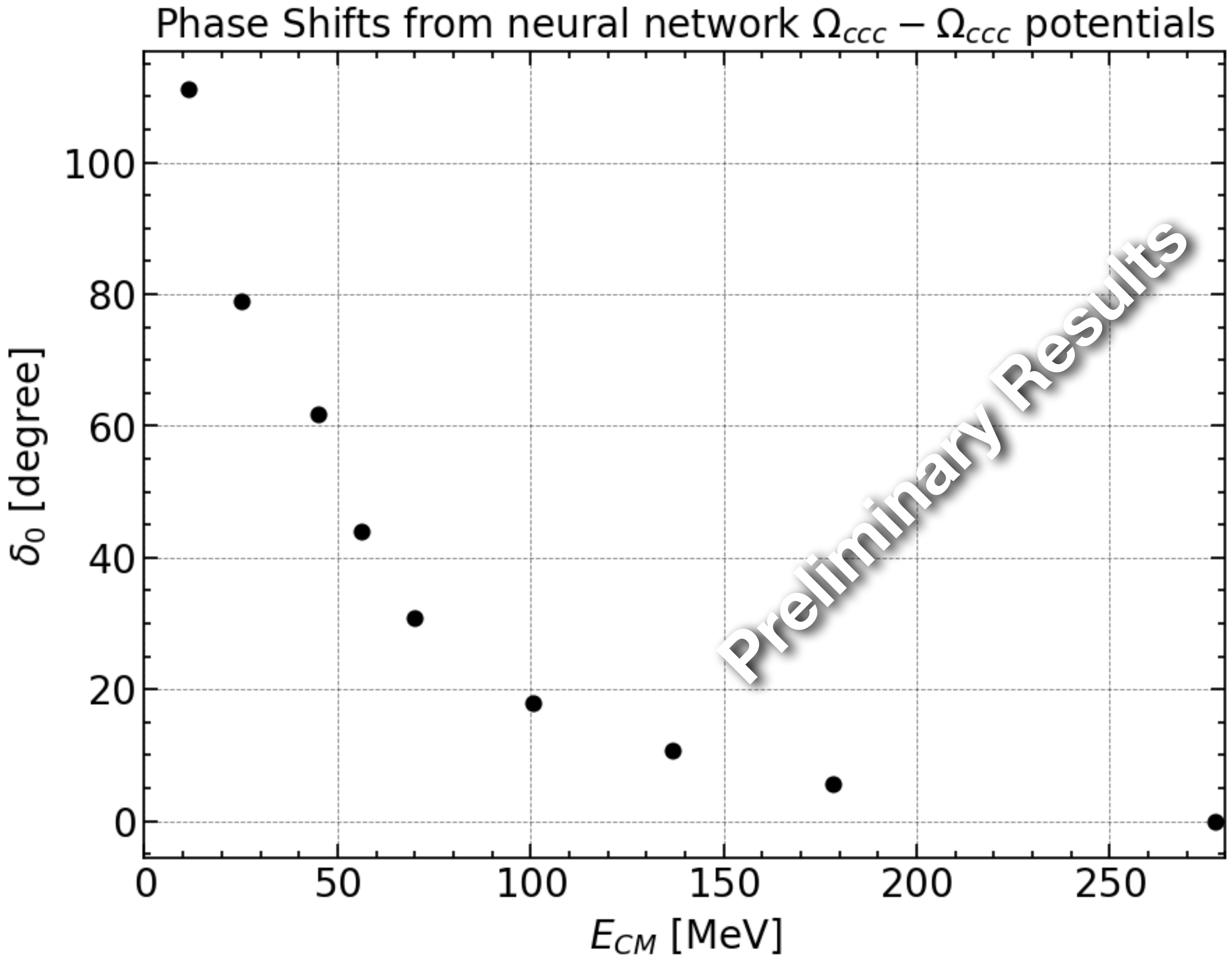
$\tilde{U}_\theta(r, r') \equiv 4\pi r'^2 U_\theta(r, r')$

Scattering Behaviors

in Preparation

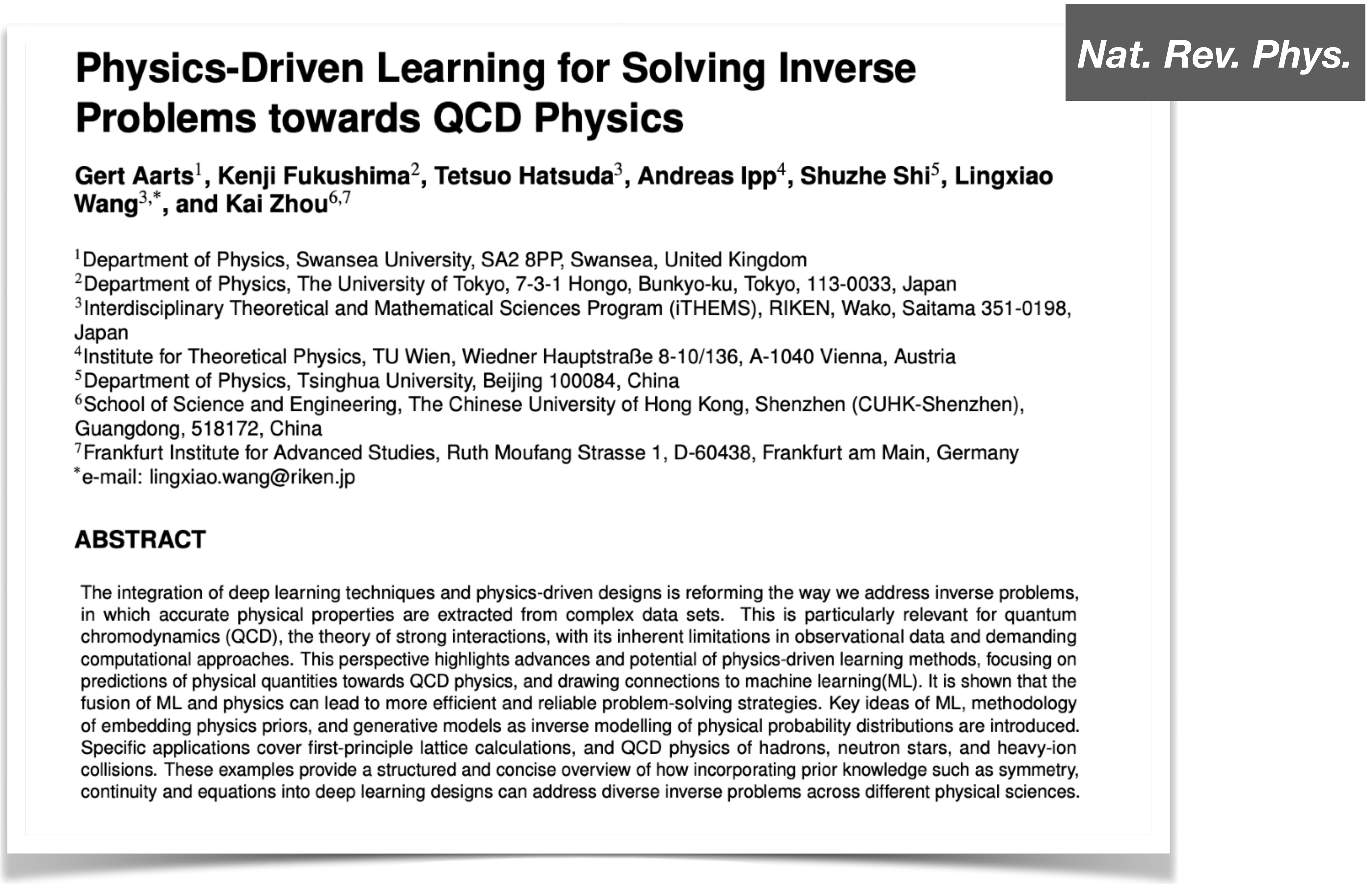
$a_0 = 0.86\,\mathrm{fm}, r_{\mathrm{eff}} = 0.57\,\mathrm{fm}$





Summary I

- **Physics-Driven Desgins**
  - Symmetry
  - Physics Principles
  - Physical Data
  - Physical Equations
- **Inverse Problems**
  - Data-Driven Learning
    - Limited data set
  - **Physics-Driven Deep Learning**
    - Neural network representations
    - Embed physics priors explicitly
      - **Exchange Symmetry** and **Asymptotic Behavior** for HH interactions



Nat. Rev. Phys.

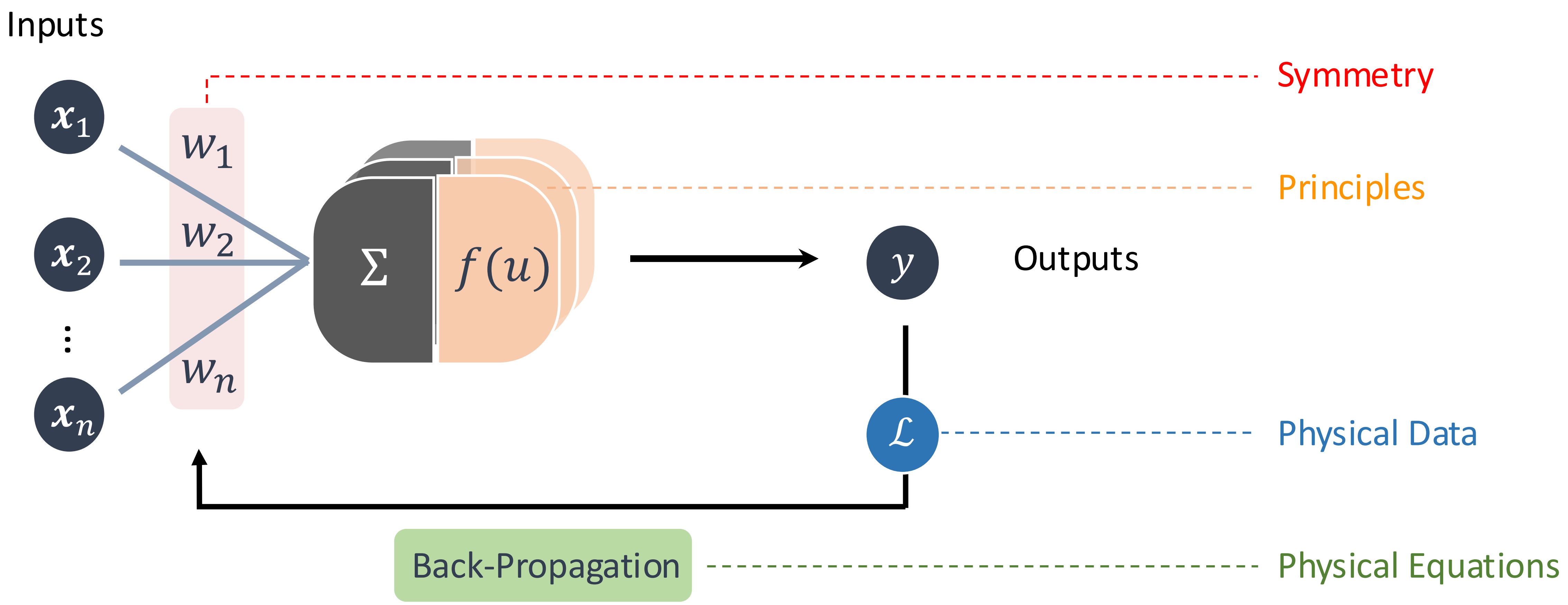
Physics-Driven Learning for Solving Inverse Problems towards QCD Physics

Gert Aarts[1], Kenji Fukushima[2], Tetsuo Hatsuda[3], Andreas Ipp[4], Shuzhe Shi[5], Lingxiao Wang[3,*], and Kai Zhou[6,7]

[1]Department of Physics, Swansea University, SA2 8PP, Swansea, United Kingdom
[2]Department of Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
[3]Interdisciplinary Theoretical and Mathematical Sciences Program (iTHEMS), RIKEN, Wako, Saitama 351-0198, Japan
[4]Institute for Theoretical Physics, TU Wien, Wiedner Hauptstraße 8-10/136, A-1040 Vienna, Austria
[5]Department of Physics, Tsinghua University, Beijing 100084, China
[6]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), Guangdong, 518172, China
[7]Frankfurt Institute for Advanced Studies, Ruth Moufang Strasse 1, D-60438, Frankfurt am Main, Germany
*e-mail: lingxiao.wang@riken.jp

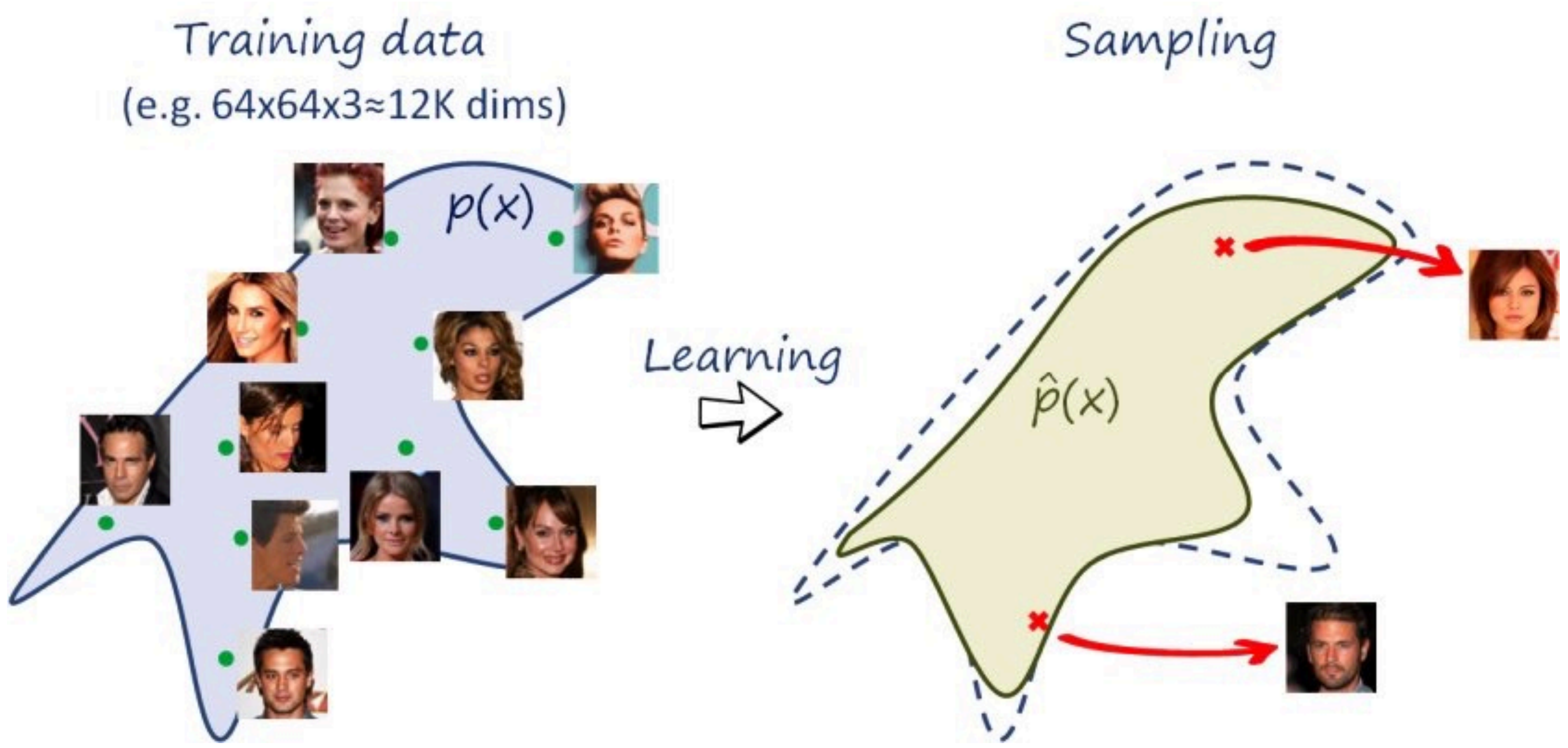
ABSTRACT

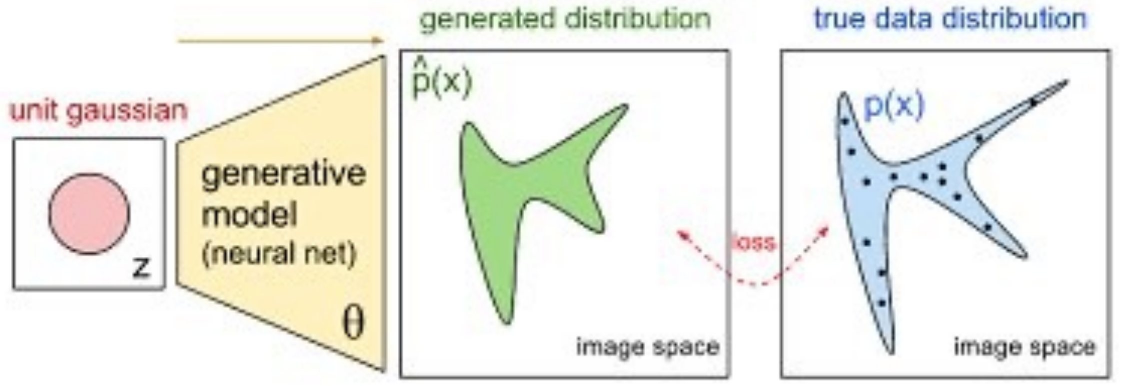
The integration of deep learning techniques and physics-driven designs is reforming the way we address inverse problems, in which accurate physical properties are extracted from complex data sets. This is particularly relevant for quantum chromodynamics (QCD), the theory of strong interactions, with its inherent limitations in observational data and demanding computational approaches. This perspective highlights advances and potential of physics-driven learning methods, focusing on predictions of physical quantities towards QCD physics, and drawing connections to machine learning(ML). It is shown that the fusion of ML and physics can lead to more efficient and reliable problem-solving strategies. Key ideas of ML, methodology of embedding physics priors, and generative models as inverse modelling of physical probability distributions are introduced. Specific applications cover first-principle lattice calculations, and QCD physics of hadrons, neutron stars, and heavy-ion collisions. These examples provide a structured and concise overview of how incorporating prior knowledge such as symmetry, continuity and equations into deep learning designs can address diverse inverse problems across different physical sciences.



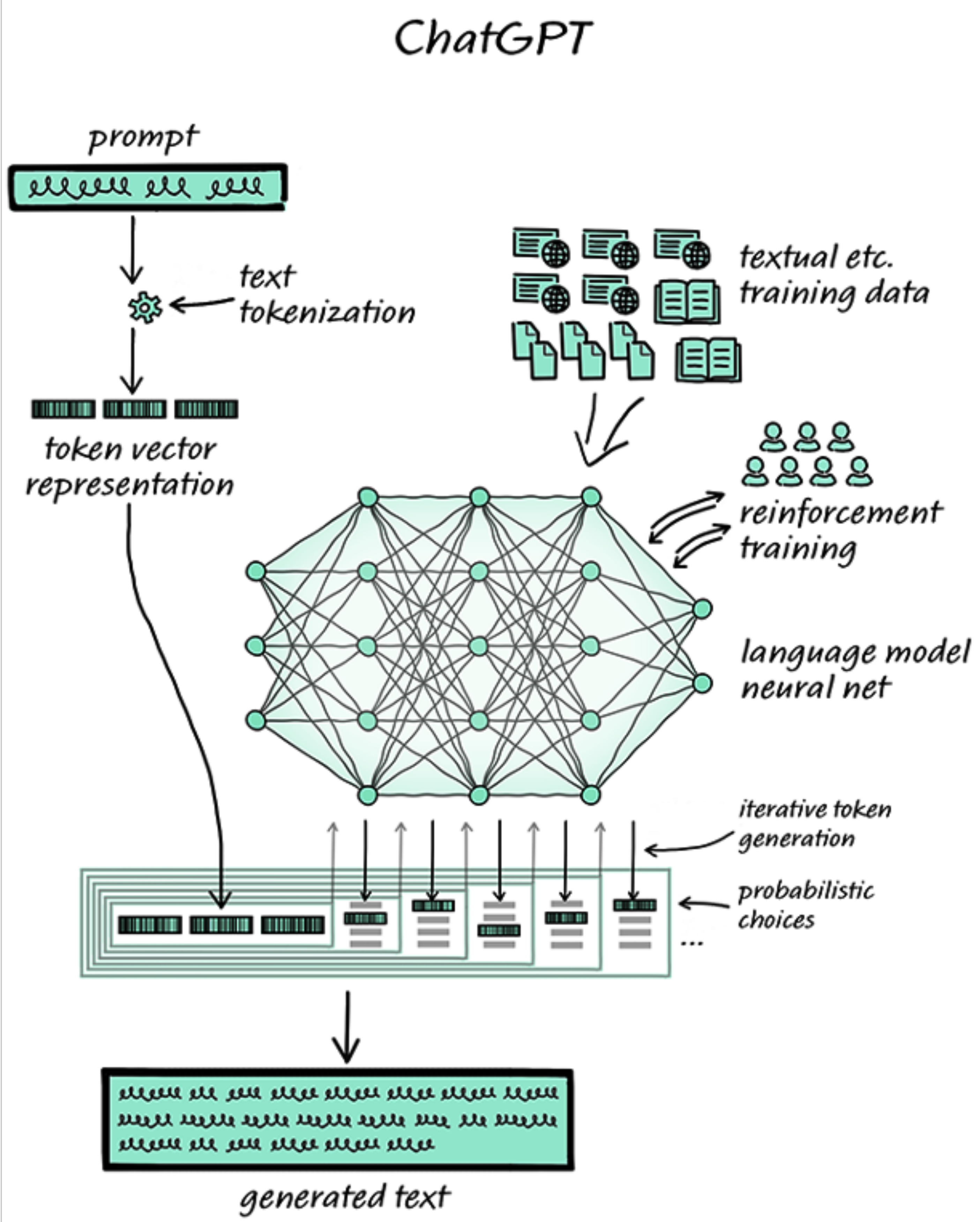
Generative Models as Inverse Modeling

Generative Models





@blogs of OpenAI



Generative models

→ **Underlying Distributions in Data**

$$\max_{\theta} \prod_{i=1}^{N} p_\theta(\mathbf{x}_i)$$

High-Dimensional Distribution

$$p(\phi) = e^{-S(\phi)}/Z$$

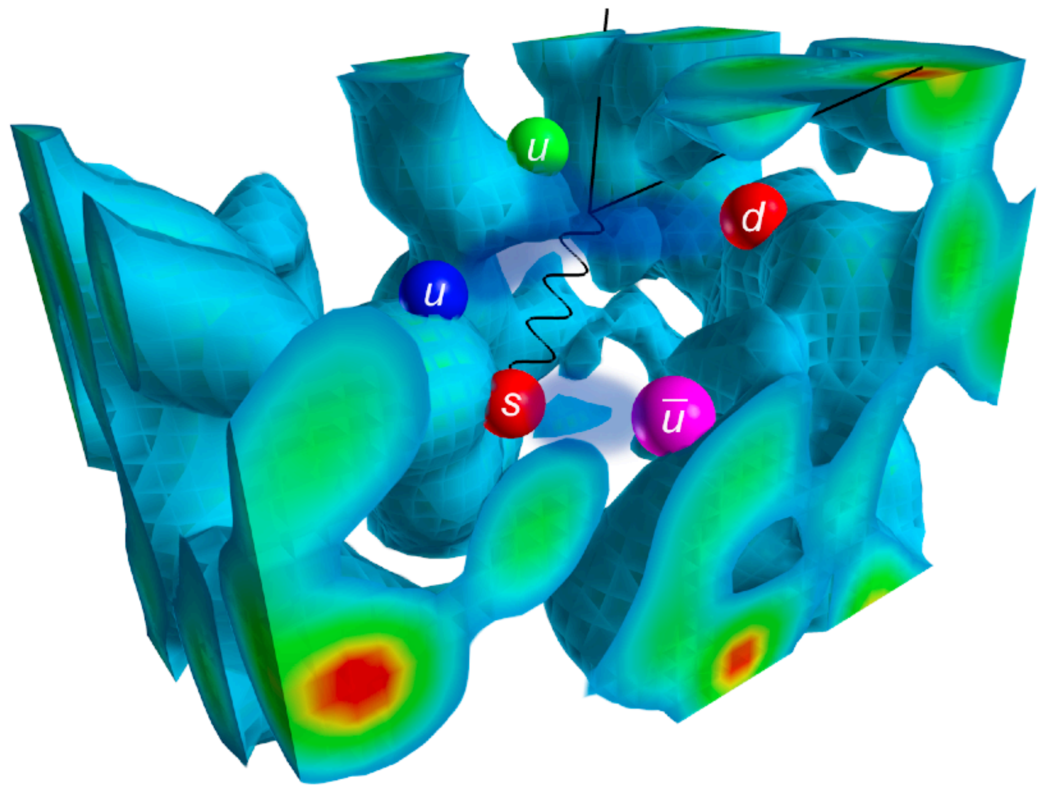
$$\langle O \rangle \approx \frac{1}{N} \sum_i O_i$$

→ **Physical Distribution, Sampling**
via Generative Models

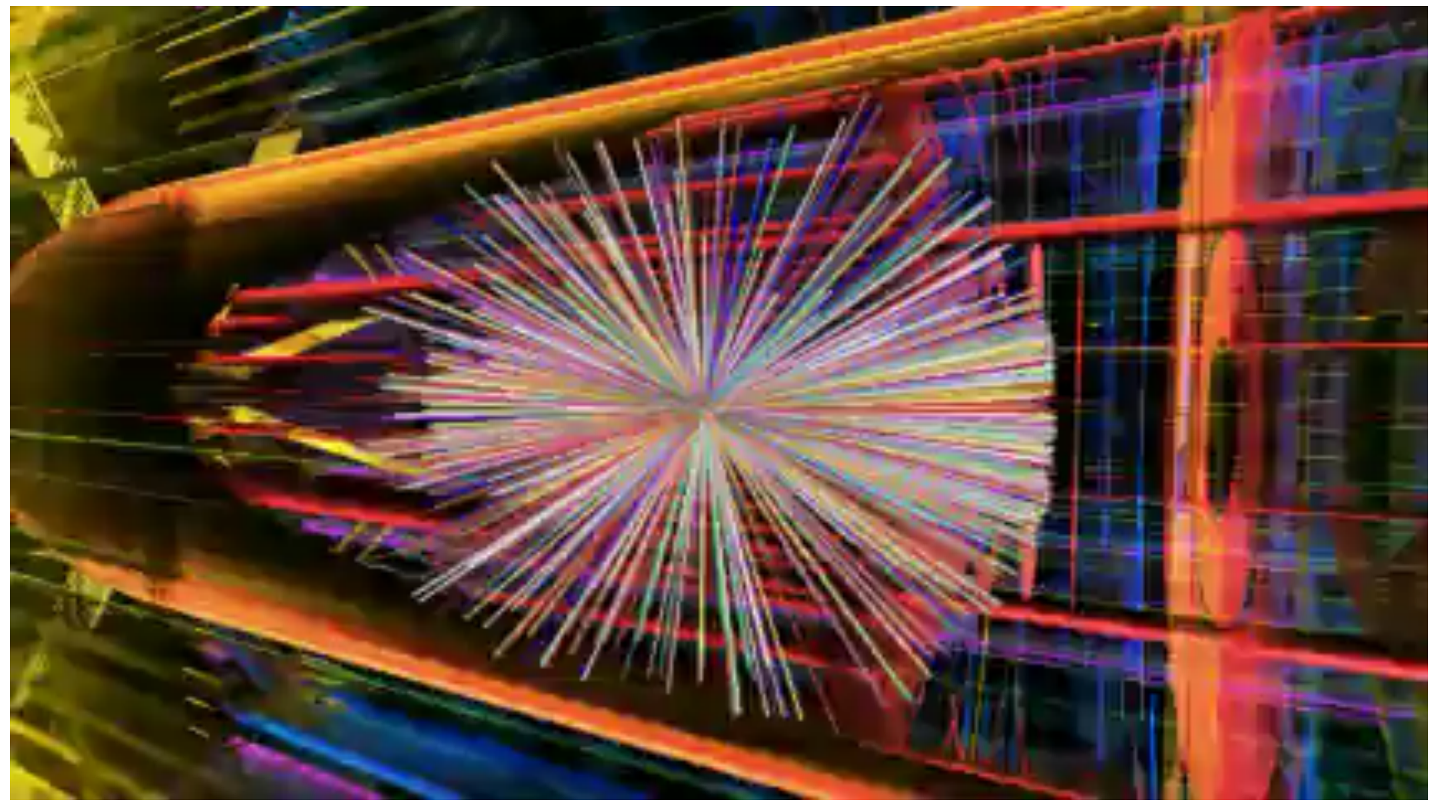
Global Sampling

Fast and Independent Sampler

Prog.Part.Nucl.Phys. 104084(2023)



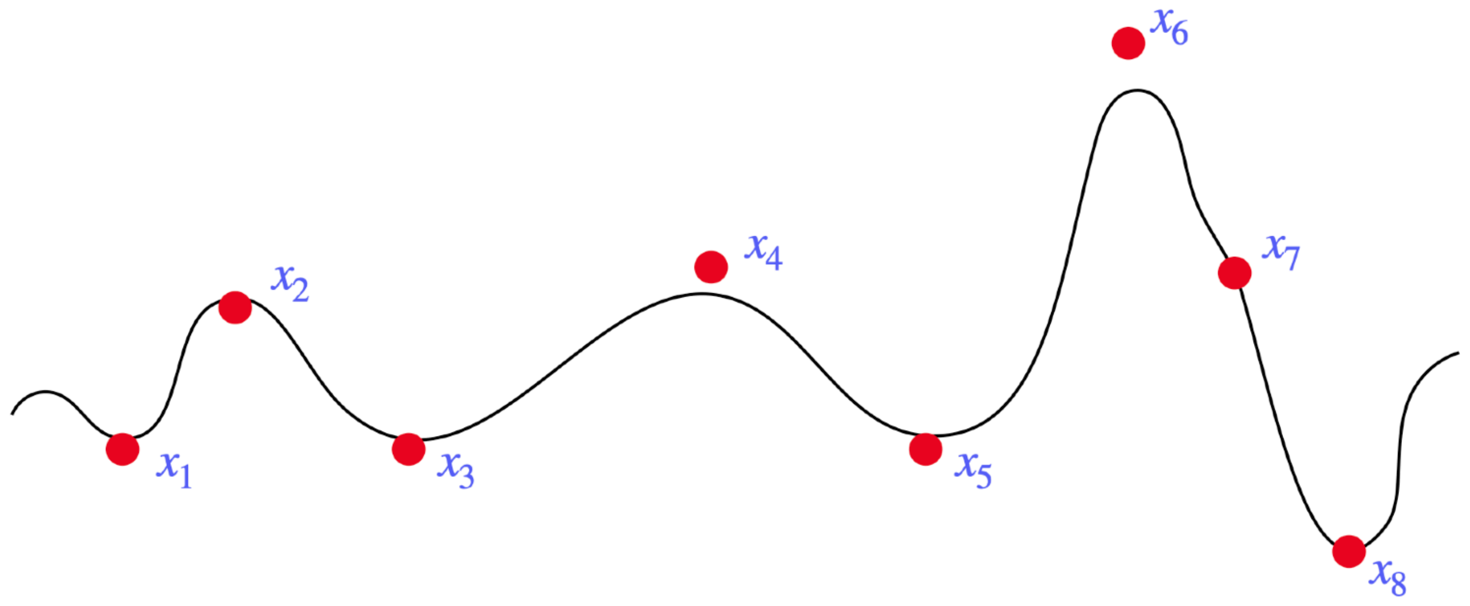
Lattice QCD © Derek Leinweber/CSSM/University of Adelaide



Heavy-Ion Collisions © 2010 CERN

Probabilistic Models

$$\mathbf{x} \sim p_{data}(\mathbf{x})$$



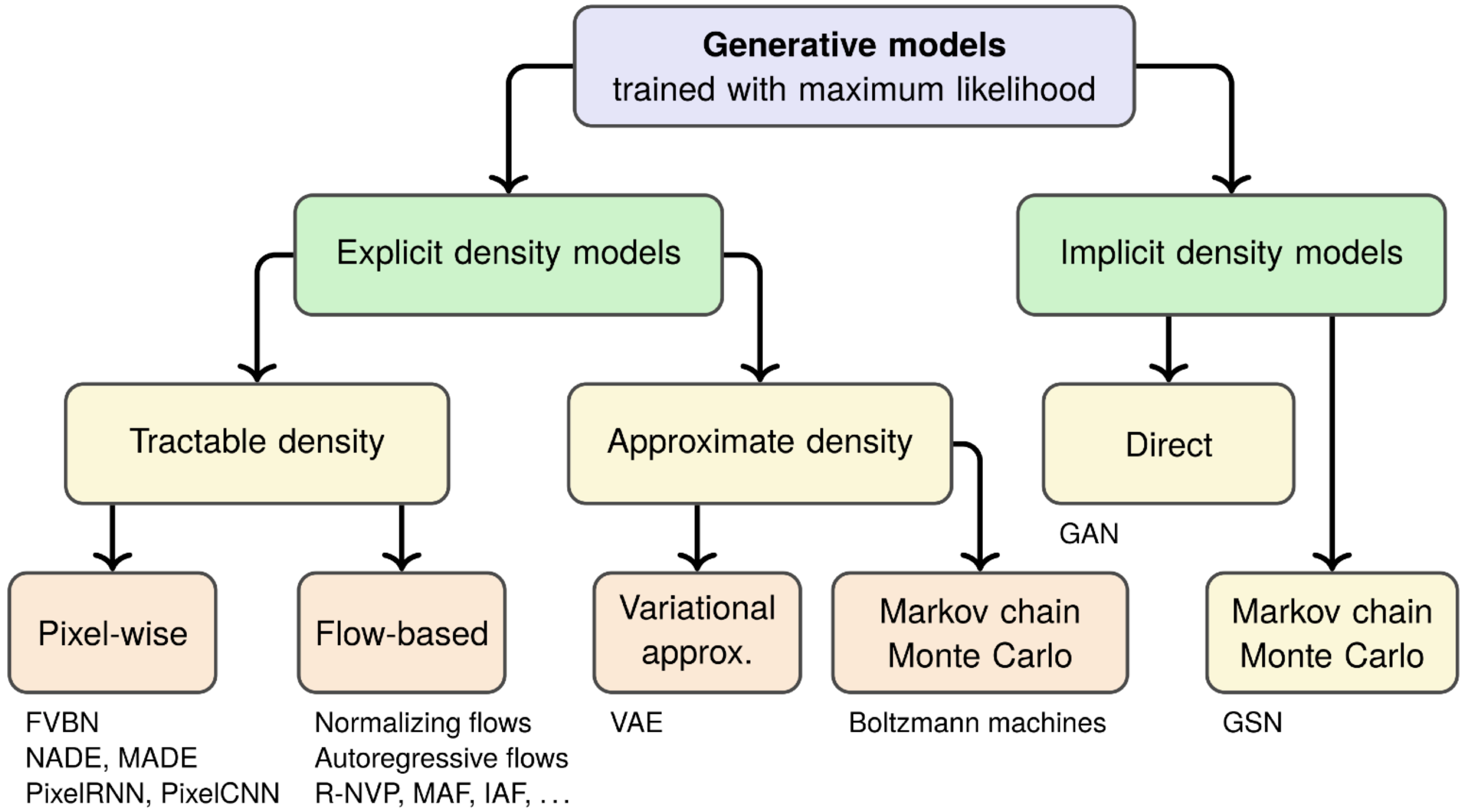
$$\max_{\theta} \prod_{i=1}^{N} p_\theta(\mathbf{x}_i)$$

Maximum likelihood estimation(MLE)

$$\max_{\theta} \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i)$$

Maximum log-likelihood estimation

Bishop, C. M. & Bishop, H. *Deep Learning: Foundations and Concepts*. doi:10.1007/978-3-031-45468-4.



I. Goodfellow, arXiv:1701.00160 (2017)

Learn to Sample

$$p(\phi) = e^{-S(\phi)}/Z$$

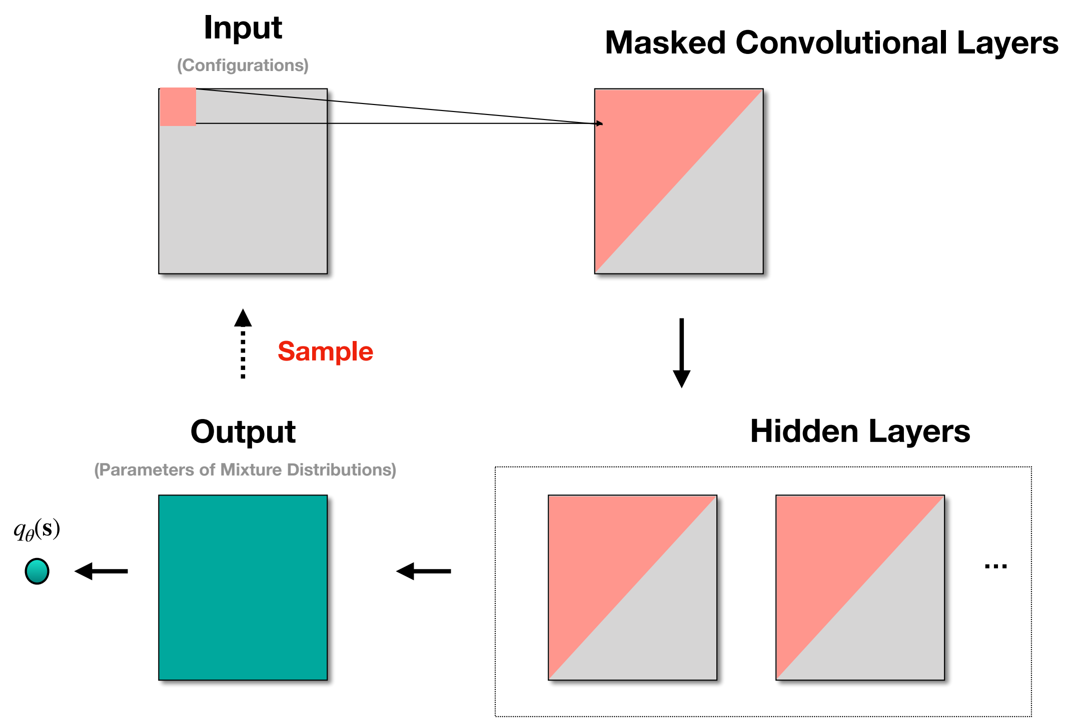
$$\langle O \rangle \approx \frac{1}{N}\sum_i O_i$$

→ **Physical Distribution, Sampling**
via Generative Models

Global Sampling

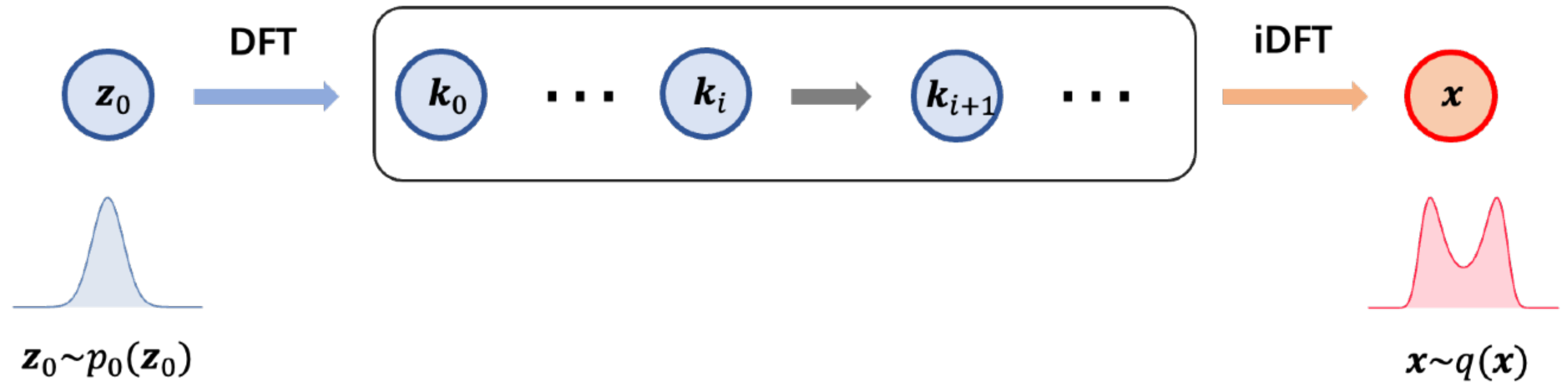
Fast and Independent Sampler

Prog.Part.Nucl.Phys. 104084(2023)



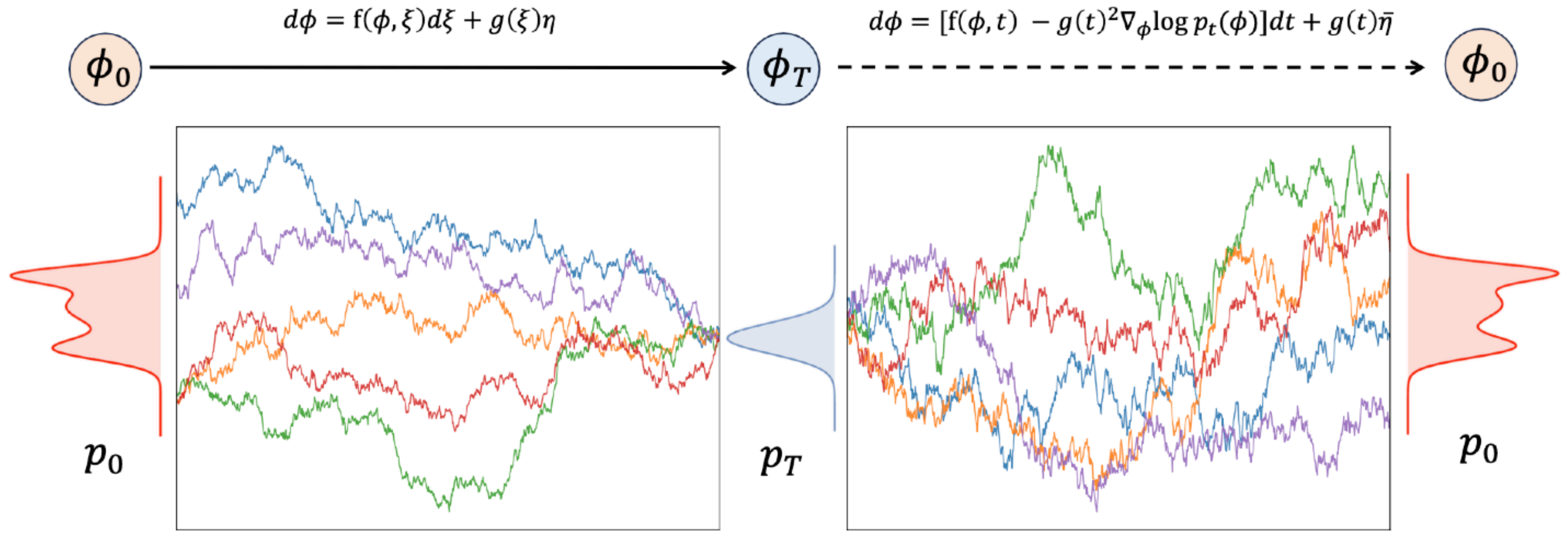
Continuous Autoregressive Models

Chinese Phys. Lett. 39, 120502 (2022)



Fourier Flow-based Model

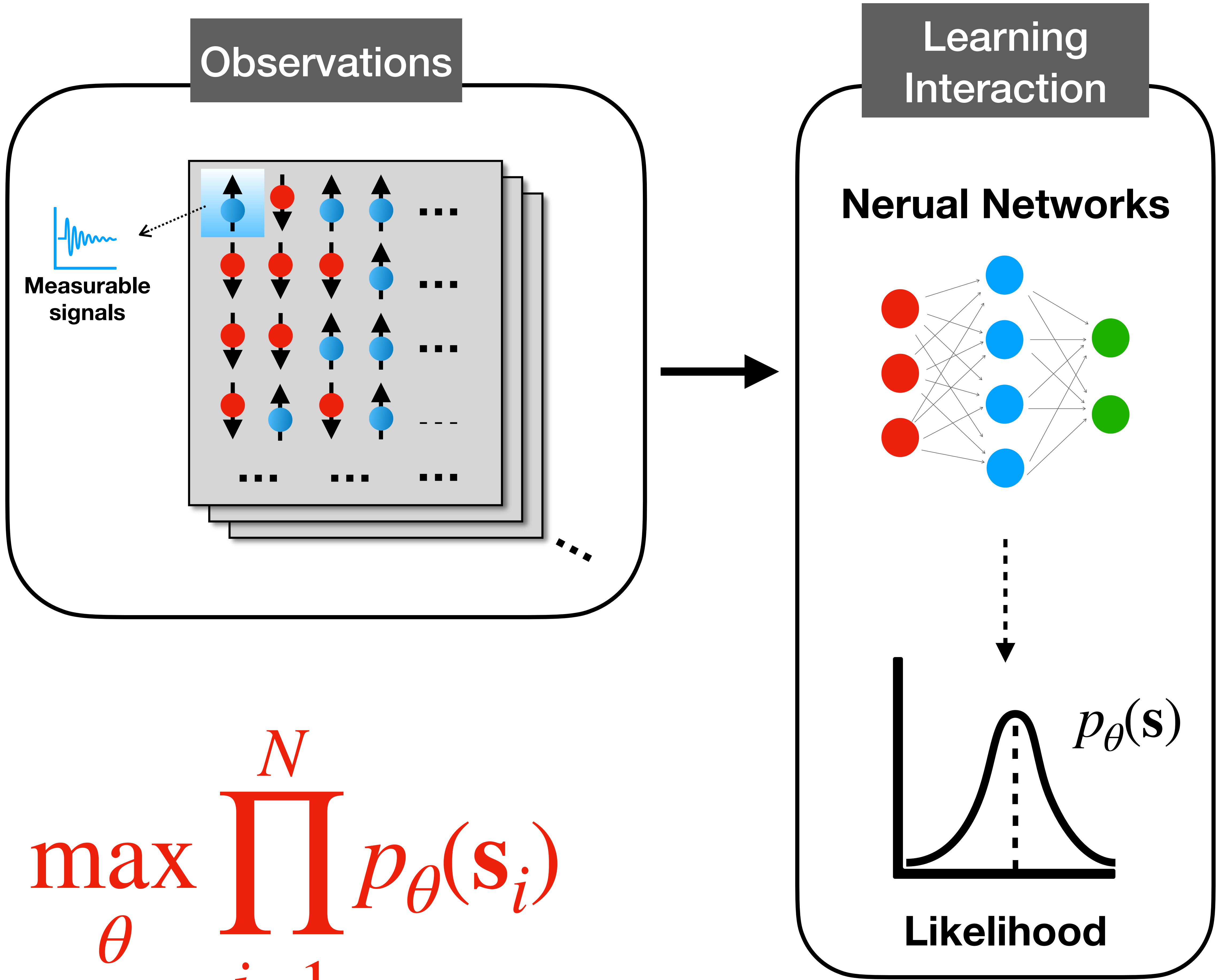
Phys. Rev. D 107, 056001



Diffusion Models

JHEP 05(2024)060

Learn Micro-Interactions from Observations



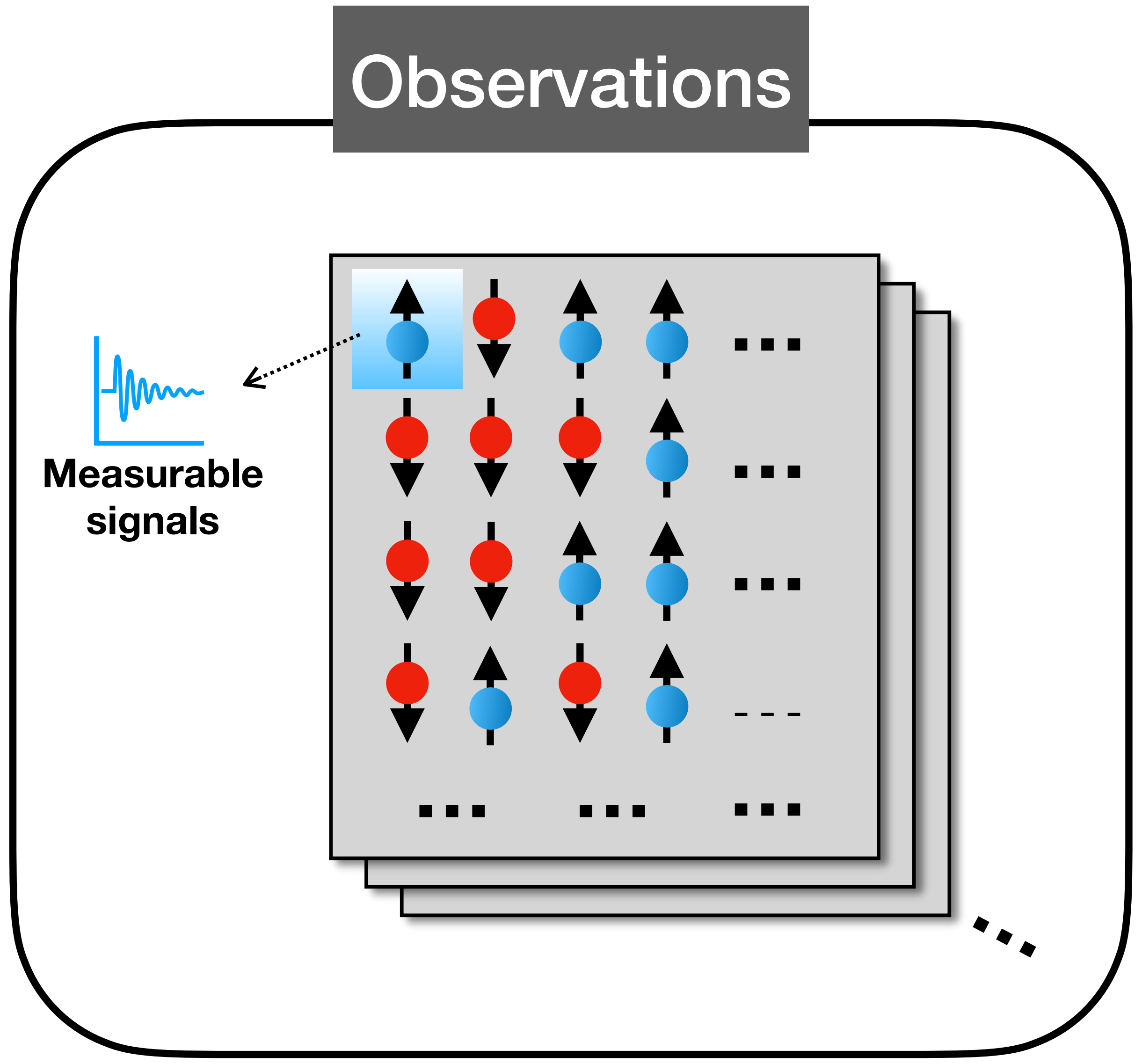
$$\max_{\theta} \prod_{i=1}^{N} p_\theta(\mathbf{s}_i)$$

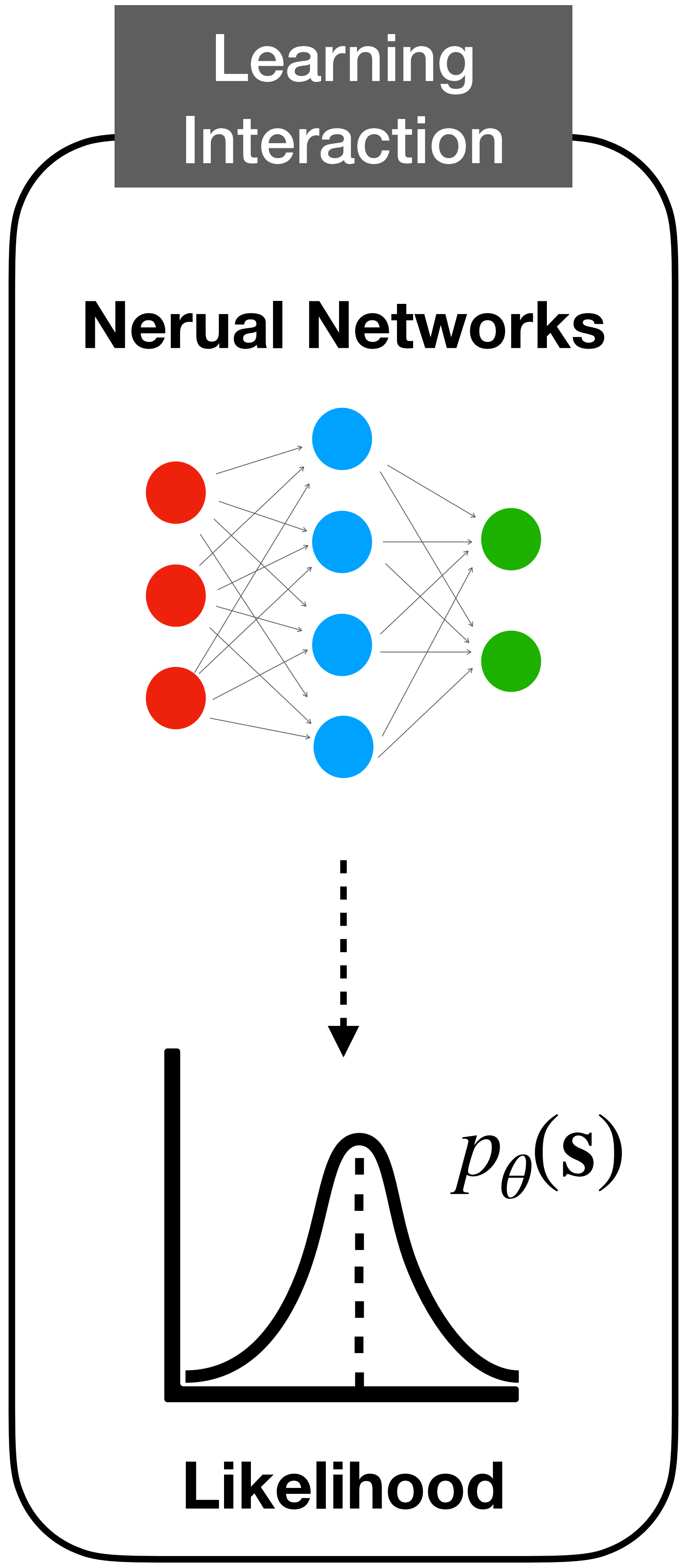
$$p(\mathbf{s}\,|\,\mathbf{T}) \leftarrow p_\theta(\mathbf{s}\,|\,\mathbf{T})$$

Learn at one specific condition(T)

Detecting phase transitions for all ?

Learn Micro-Interactions from Observations





Learn at one specific condition(T)

Detecting phase transitions for all

$$\max_{\theta} \prod_{i=1}^{N} p_\theta(\mathbf{s}_i)$$

$$p(\mathbf{s}) \leftarrow p_\theta(\mathbf{s}) \equiv \frac{e^{-\frac{H_\theta(\mathbf{s})}{T}}}{Z}$$

$$H_\theta(\mathbf{s}, T) = -T \ln p_\theta(\mathbf{s}) - T \ln Z$$

$$\frac{\Delta H_\theta(\mathbf{s}, T)}{T'} \equiv -\frac{T}{T'}(\ln p_\theta(\mathbf{s} + \delta \mathbf{s}) - \ln p_\theta(\mathbf{s}))$$

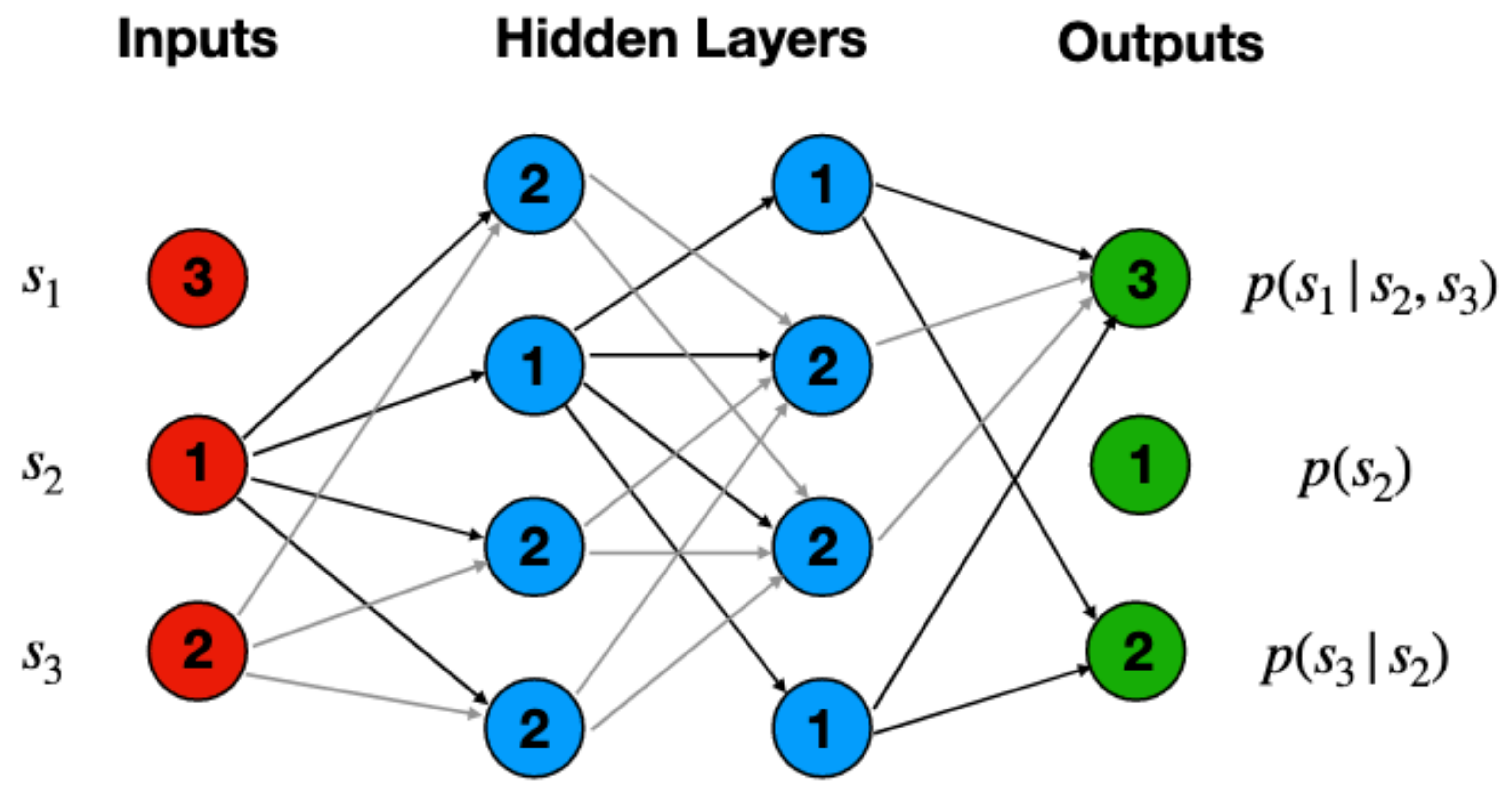
Autoregressive Networks

$$p_\theta(s) = \prod_{i=1}^{N} p(s_i \,|\, s_1, \ldots, s_{i-1})$$

- **Example (L=3) for 1D spin model**

$$p_\theta(s) = p(s_3 \,|\, s_2, s_1) p(s_2 \,|\, s_1) p(s_1)$$

- $p(s_i \,|\, s_{<i})$ **can be any naive distribution**



Chinese Phys. Lett. 39, 120502 (2022)

Bernoulli distribution for Ising model

$$p(s_i \,|\, s_{<i}) = q_i \delta_{s_i,+1} + (1 - q_i)\delta_{s_i,-1}$$

$q_1 = f(s_1 = +1), q_2 = f(s_2 = +1 \,|\, s_1), q_3 = f(s_3 = +1 \,|\, s_2, s_1)$

discrete *d.o.f.s*

Beta distribution for continuous *d.o.f.*, $X \sim Beta(a, b)$

$$p_\theta(s_i \,|\, s_1, \cdots, s_{i-1}) = \frac{\Gamma(a_i + b_i)}{\Gamma(a_i)\Gamma(b_i)} s_i^{a_i - 1}(1 - s_i)^{b_i - 1}$$

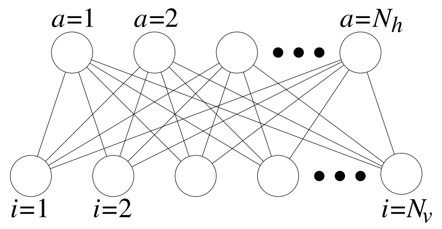
$\Gamma(a)$ is gamma function, $s_i = \theta_i / 2\pi \in [0,1), (a_i, b_i) > 0$

continuous *d.o.f.s*

Autoregressive Networks

$$p_\theta(s) = \prod_{i=1}^{N} p(s_i \,|\, s_1, \ldots, s_{i-1})$$

Network is parametrized by a triangular matrix L, which ensures that s_i is independent with s_j when $j \geq i$. This is named as autoregressive property in machine learning.



Gaussian scalar field RBM

- induced distribution on visible layer

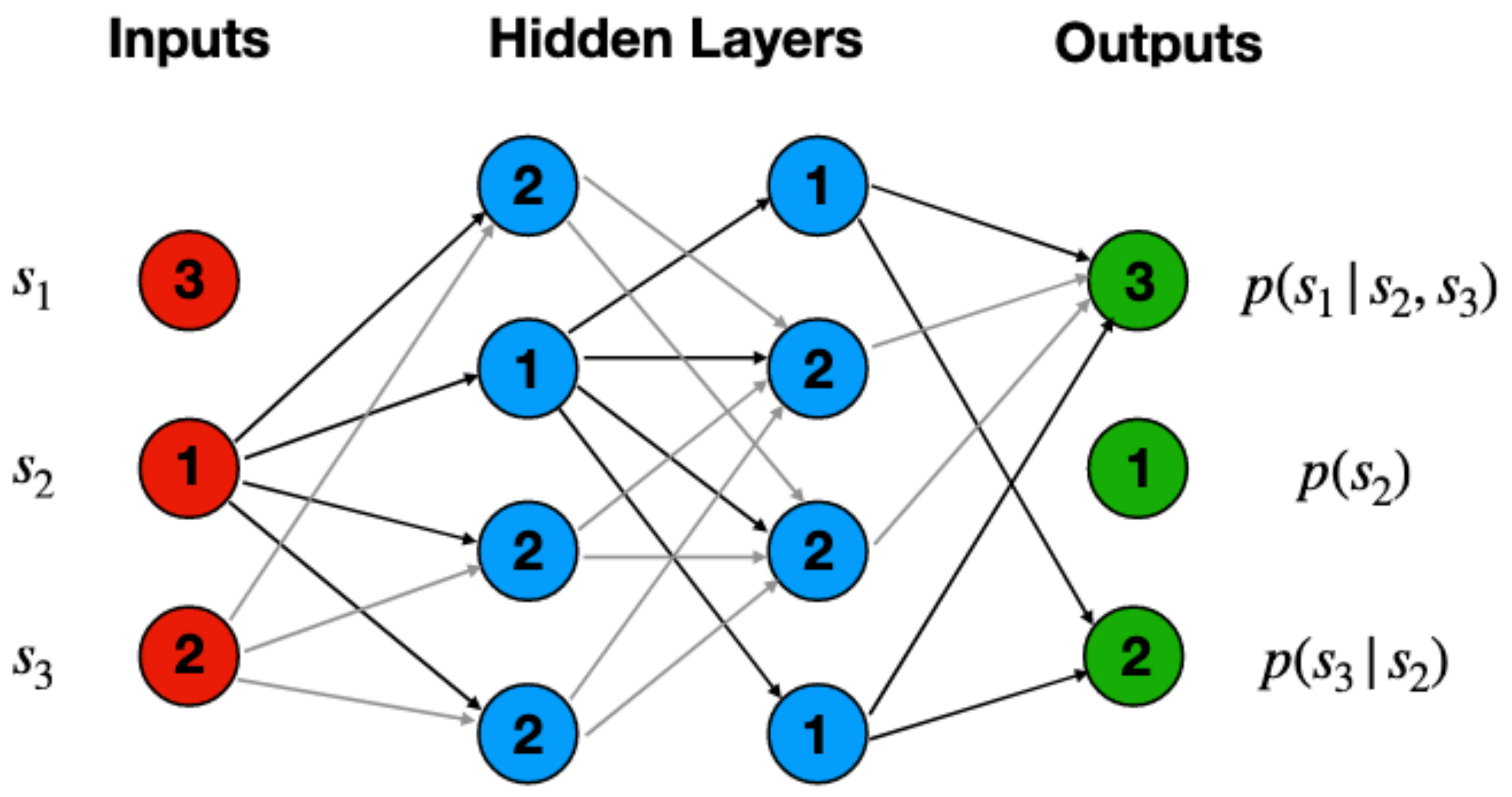
$$p(\phi) = \int Dh\, p(\phi, h) = \frac{1}{Z} \exp\left(-\frac{1}{2}\sum_{i,j} \phi_i K_{ij} \phi_j + \sum_i J_i \phi_i\right)$$

- scalar field with kinetic (all-to-all) term $K_{ij} = \mu_i^2 \delta_{ij} - \sigma^2 \sum_a w_{ia} w_{aj}^T$

  and source $J_i = \sum_a w_{ia} \eta_a$

- unusual Gaussian LFT: what is the weight matrix W and bias η?

Gert's slides@XQCD2023



$$WW^T = \frac{1}{\sigma^2}\left(\mu^2 \mathbb{1} - K^\phi\right) \equiv \mathcal{K}$$

Exact results for $N_h = N_v$

(infinitely) many solutions for weight matrix: $\mathcal{K}$ is symmetric and positive-definite

1. Cholesky decomposition $\mathcal{K} = LL^T$: $W = L$ triangular
2. diagonalisation $\mathcal{K} = ODO^T = O\sqrt{D}O^T O\sqrt{D}O^T$: $W = W^T = O\sqrt{D}O^T$
3. non-uniqueness: internal symmetry $W \to WO_R$ → $\phi^T W h \to \phi^T W O_R h = \phi^T W h'$

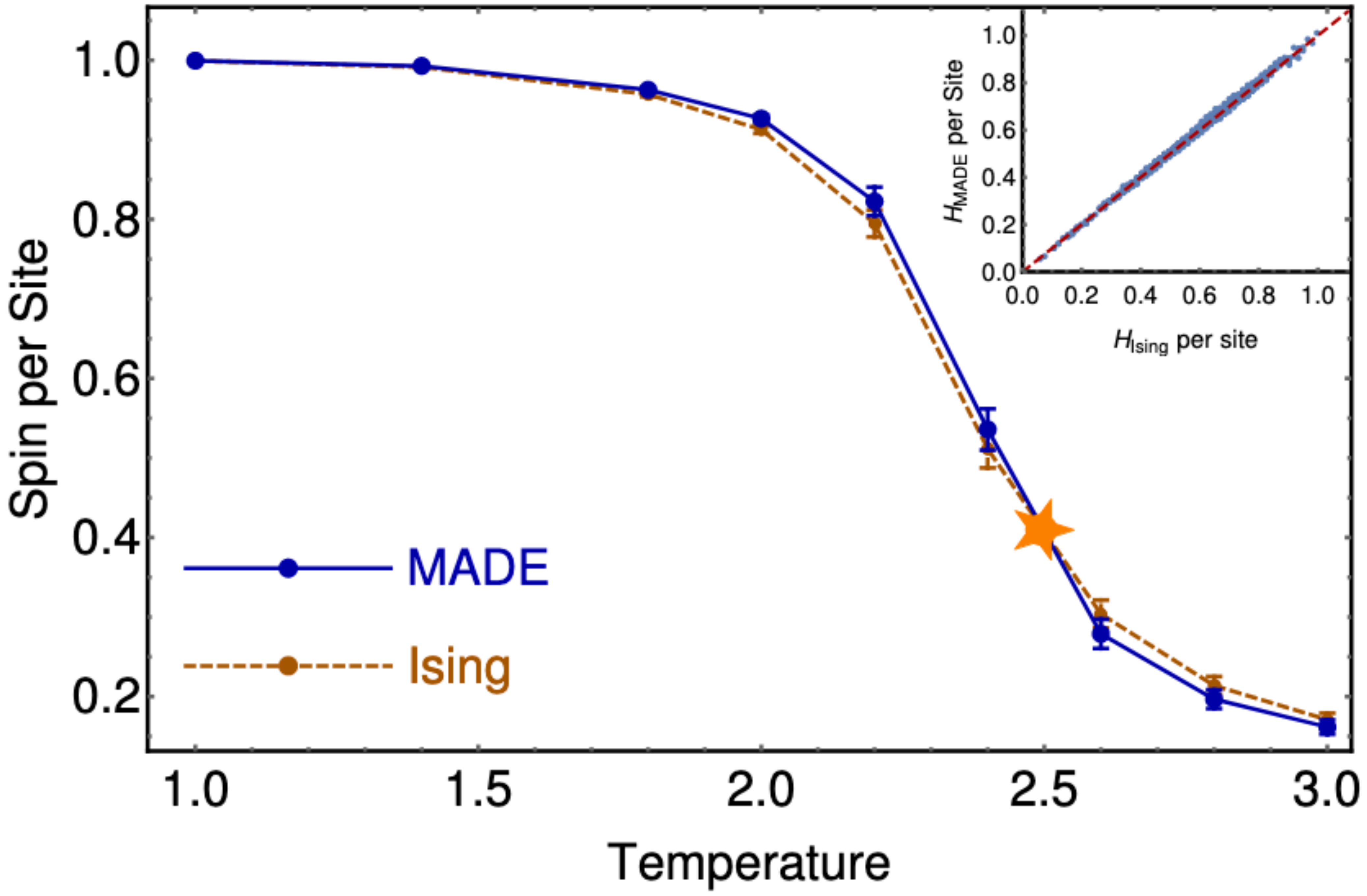
in practice

- all equally valid, realisation depends on initialisation
- non-observable degeneracy due to internal symmetry on hidden layer

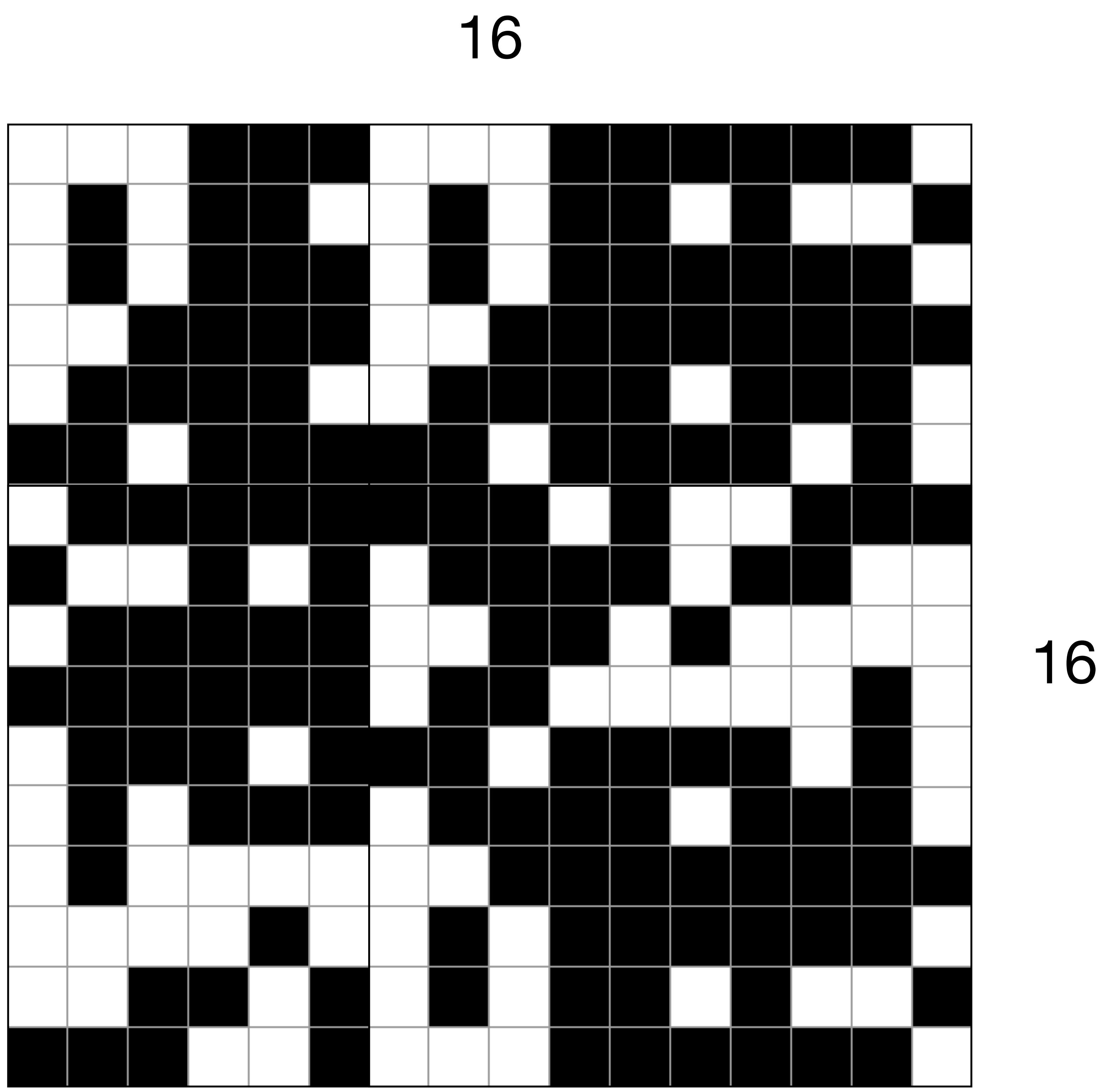
Ferromagnetic Phase Transition

2D Ising Model

arXiv:2007.01037



Masked Autoencoder for Distribution Estimation **(MADE)**

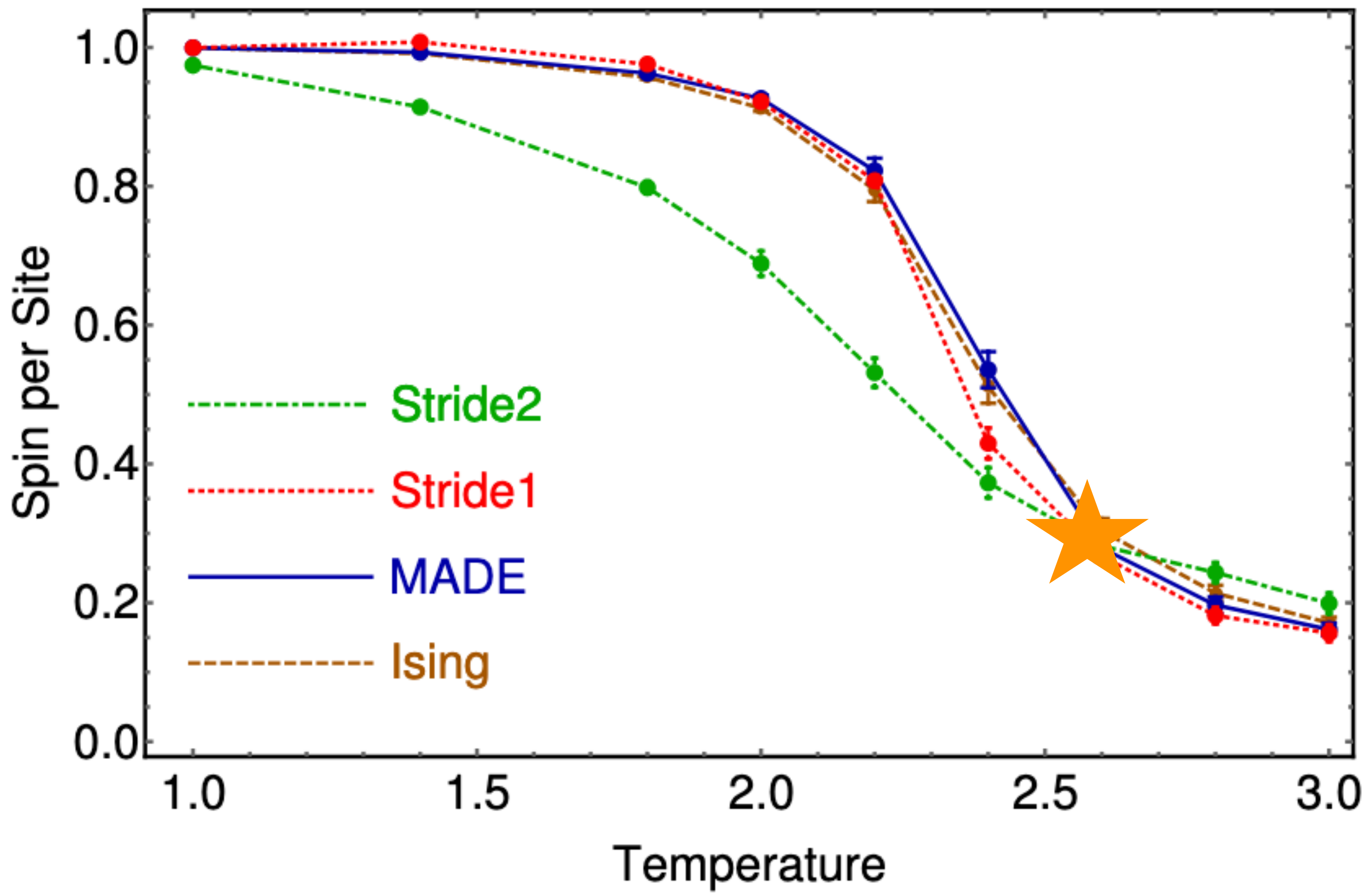


$$H(\mathbf{s}) = -\sum_{<i,j>} s_i s_j$$

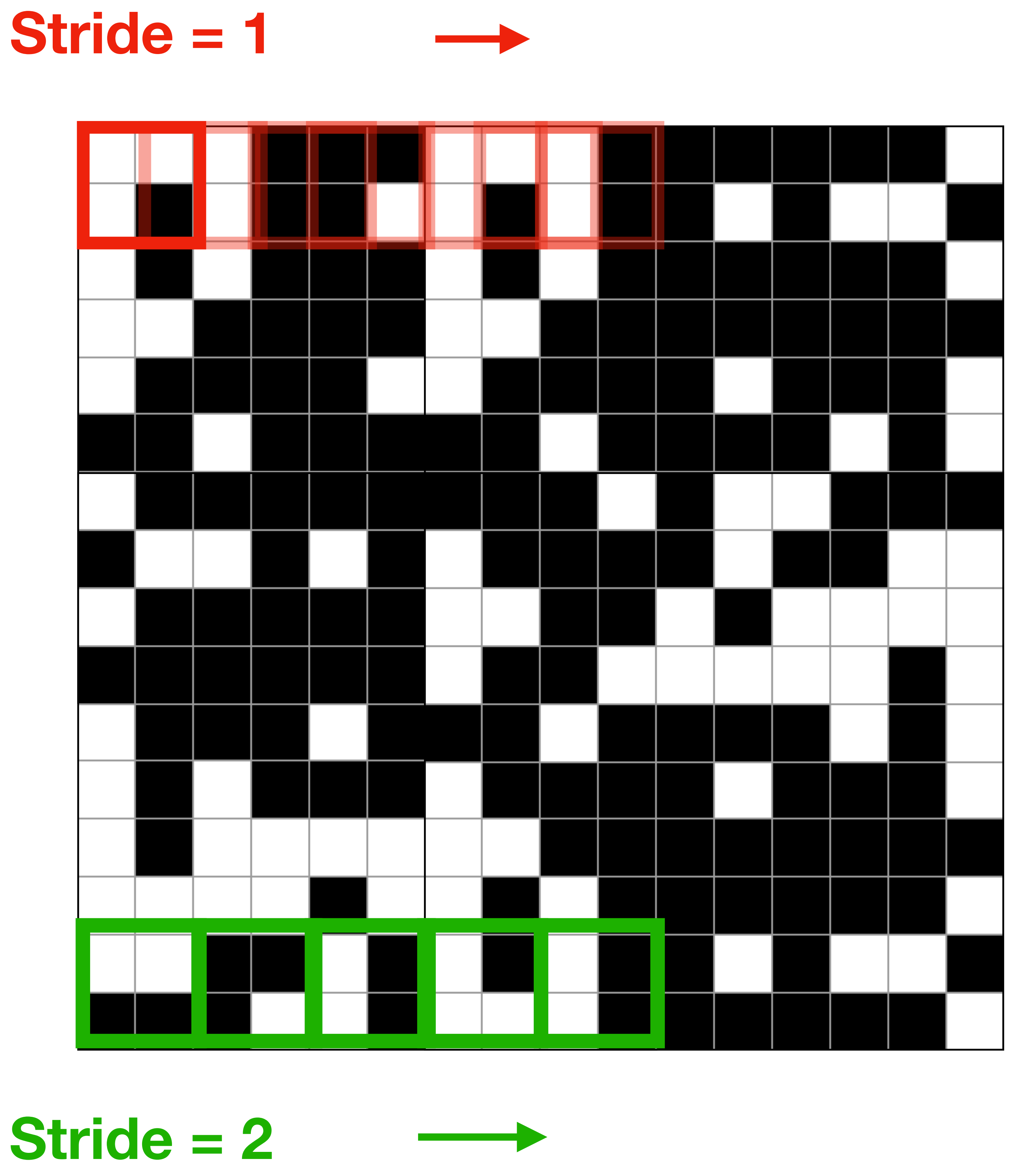
Ferromagnetic Phase Transition

2D Ising Model

arXiv:2007.01037



Masked Autoencoder for Distribution Estimation **(MADE)**

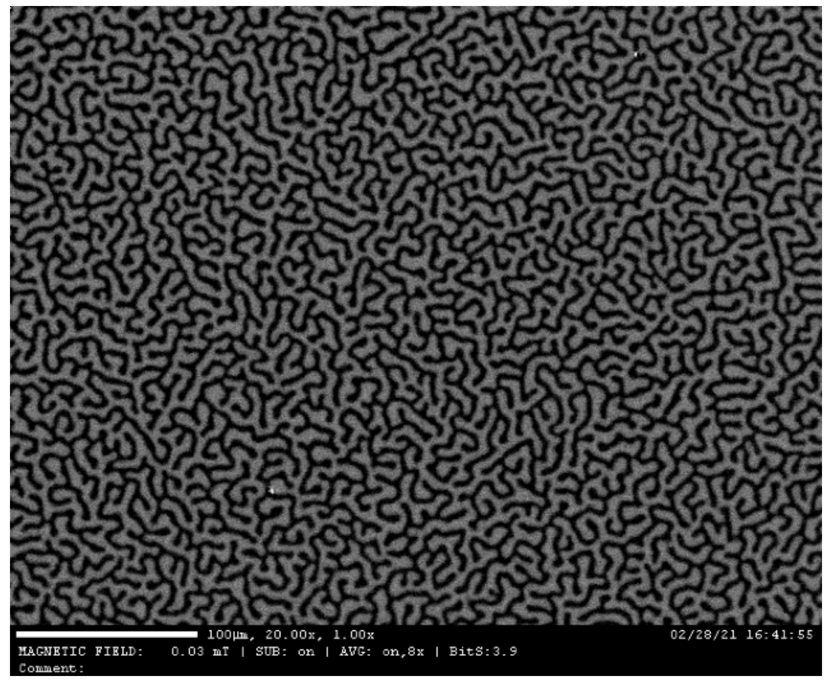


2×2 **spin-block**

Learn to Detect Phase Transitions

Ferromagnetic Materials

In preparation

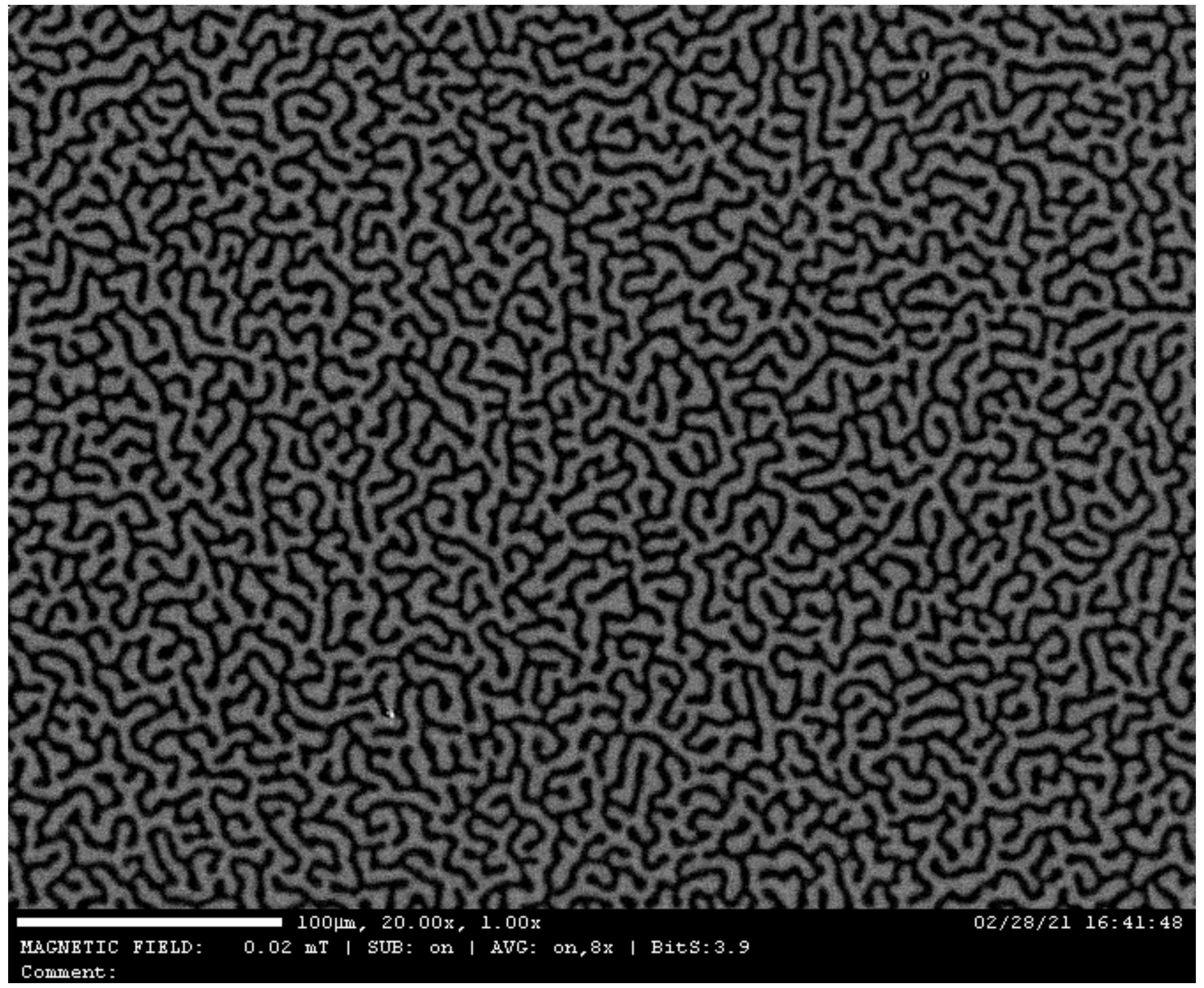


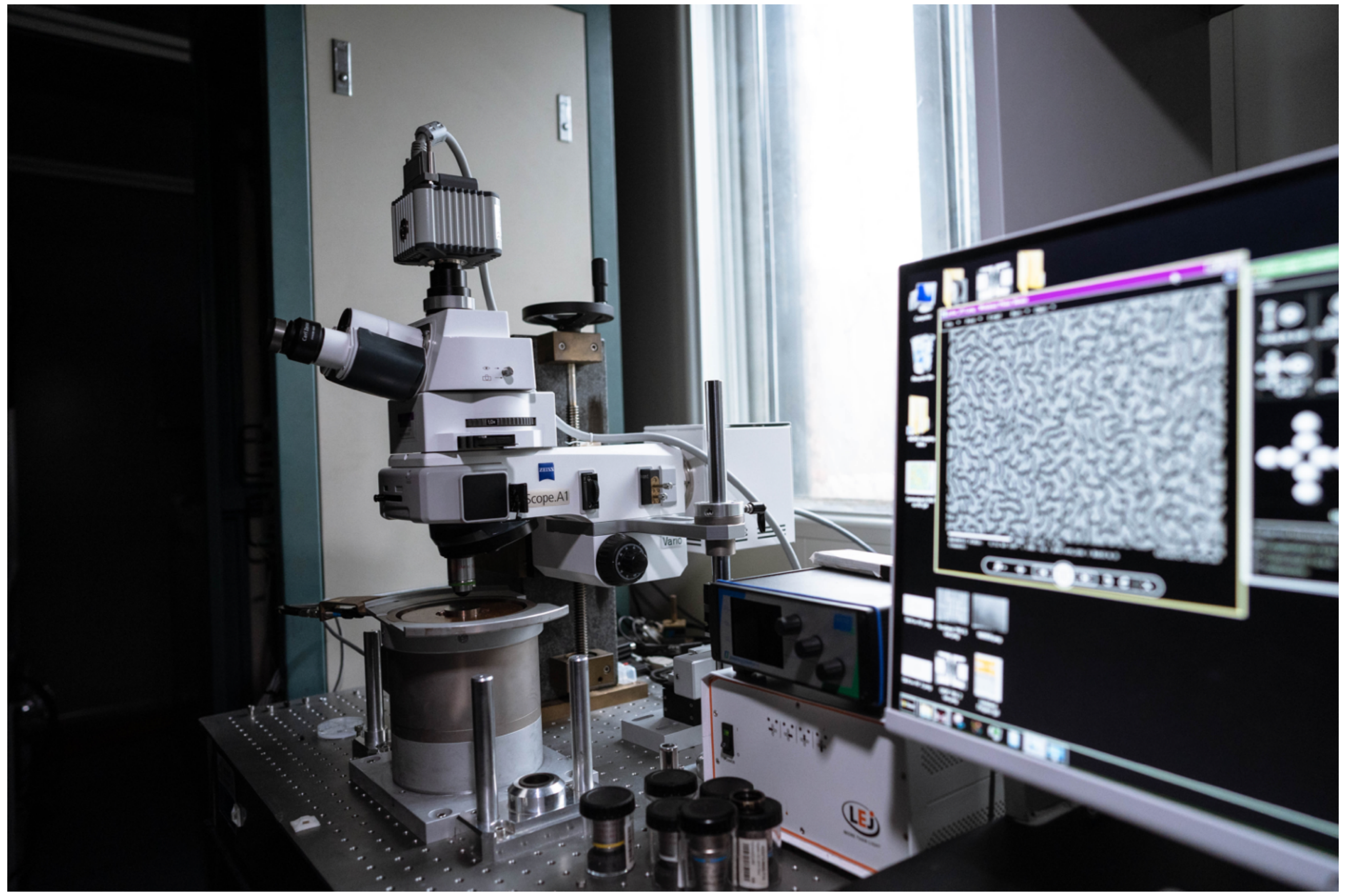
…

3000 images

Resolution: 0.5 μm

PhysRevLett.125.027206





MOKE microscope@THU

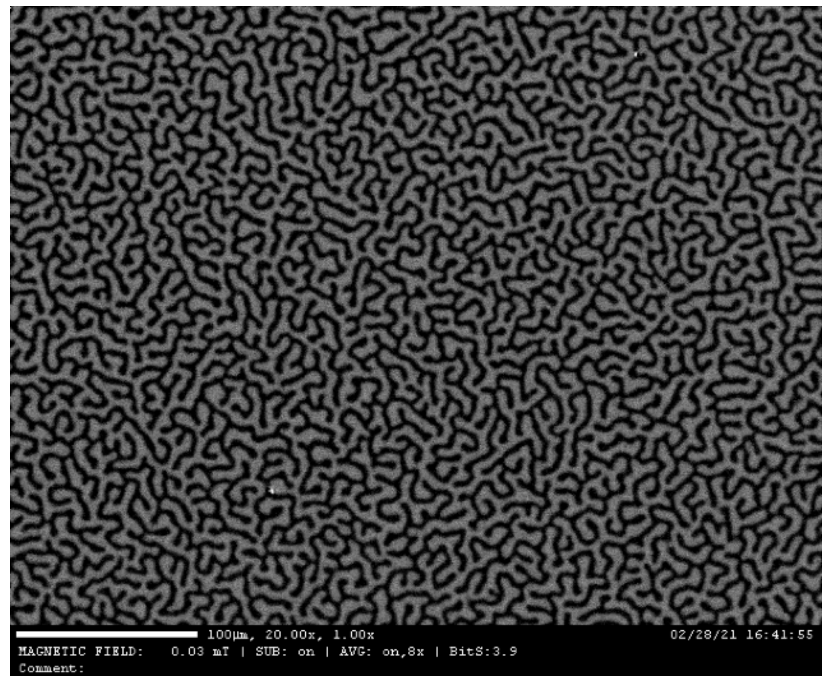
with Le Zhao and Wan-Jun Jiang

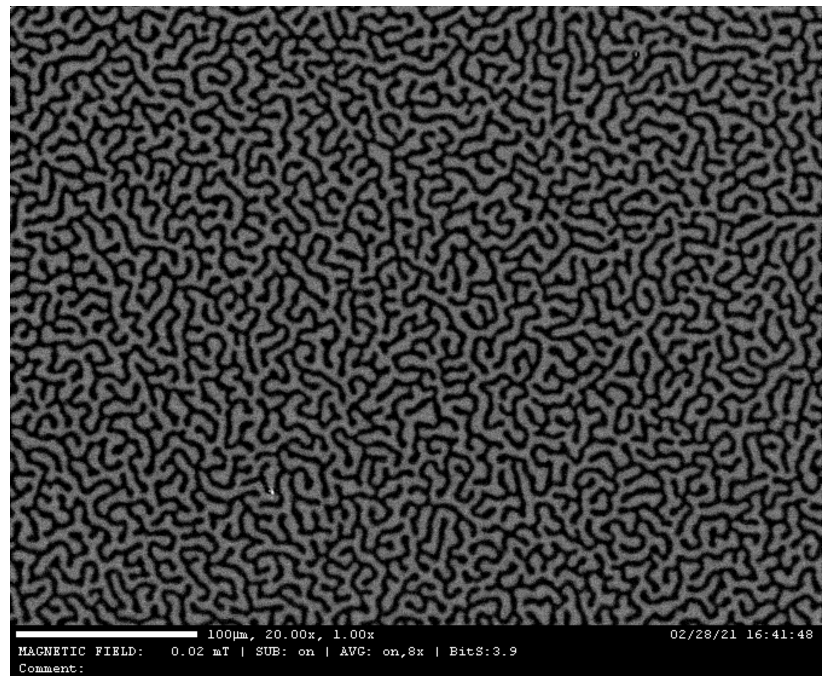
Magneto-optic Kerr effect (MOKE) microscope to capture images
for the **magnetic domains** appearing inside a **Ta/CoFeB/TaO_x thin film**
at **room temperature T = 296 K**

Learn to Detect Phase Transitions

Ferromagnetic Materials

In preparation





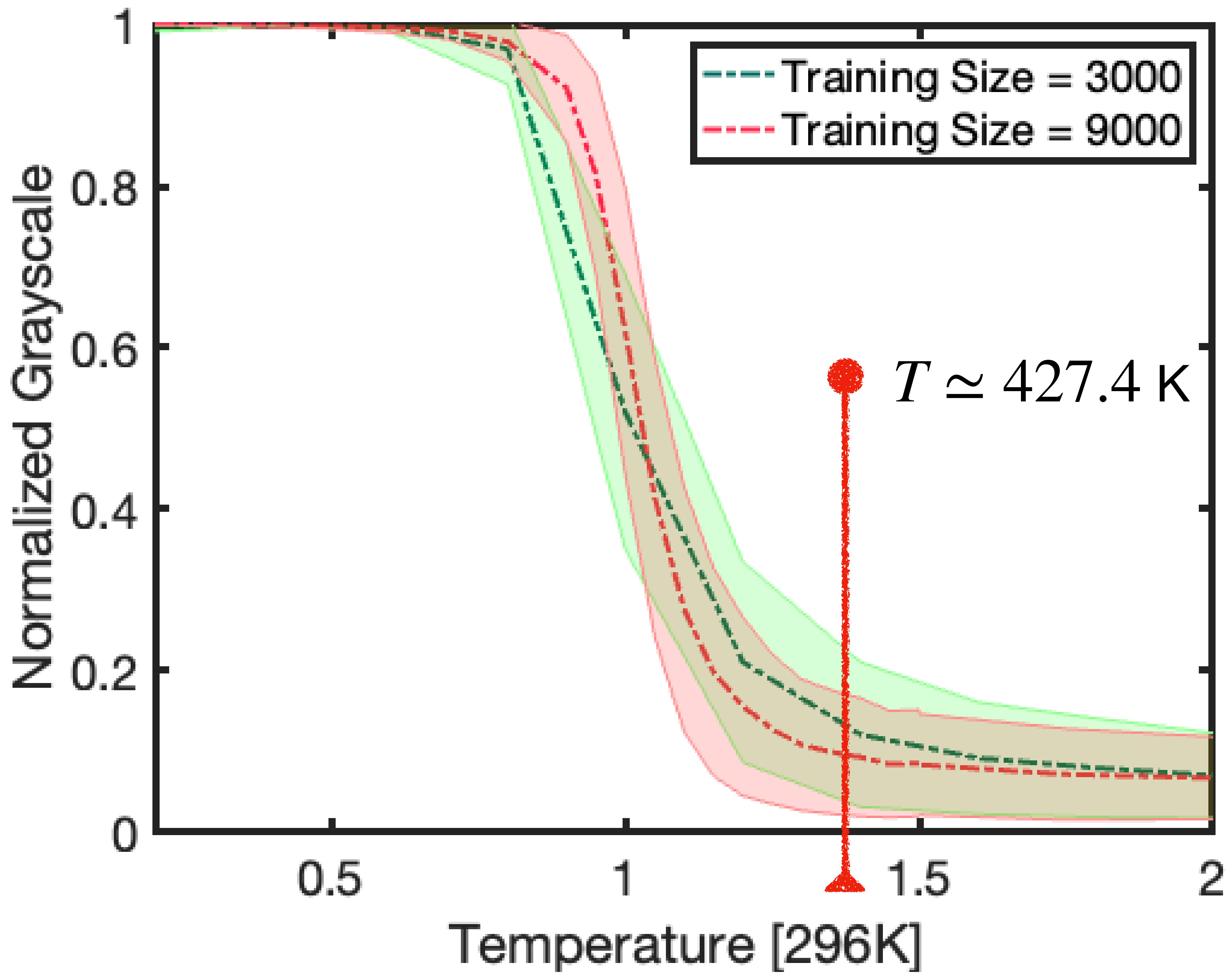
PhysRevLett.125.027206

3000 images

T [K]	50	100	150	200	250	300
M_s [emu/cc]	961.16	925.25	877.73	818.17	753.31	673.427

$$M_s(T) = M_s(0)(1 - T/T_C)^{1/3}$$

$$T_C = 427.4 \pm 2.9 \text{ K}$$



Finite-Temperature Fields

$$Z = \int D\Phi \exp(-S[\Phi])$$

$$S[\Phi] = \sum \Delta\tau(\Delta x)^3[(\frac{\Delta\Phi}{\Delta\tau})^2 + (\nabla\Phi)^2 + V(\Phi)$$

$$= \sum (\Delta x)^3 \left[\frac{(\Delta\Phi)^2}{\Delta\tau} + \Delta\tau((\nabla\Phi)^2 + V(\Phi)) \right]$$

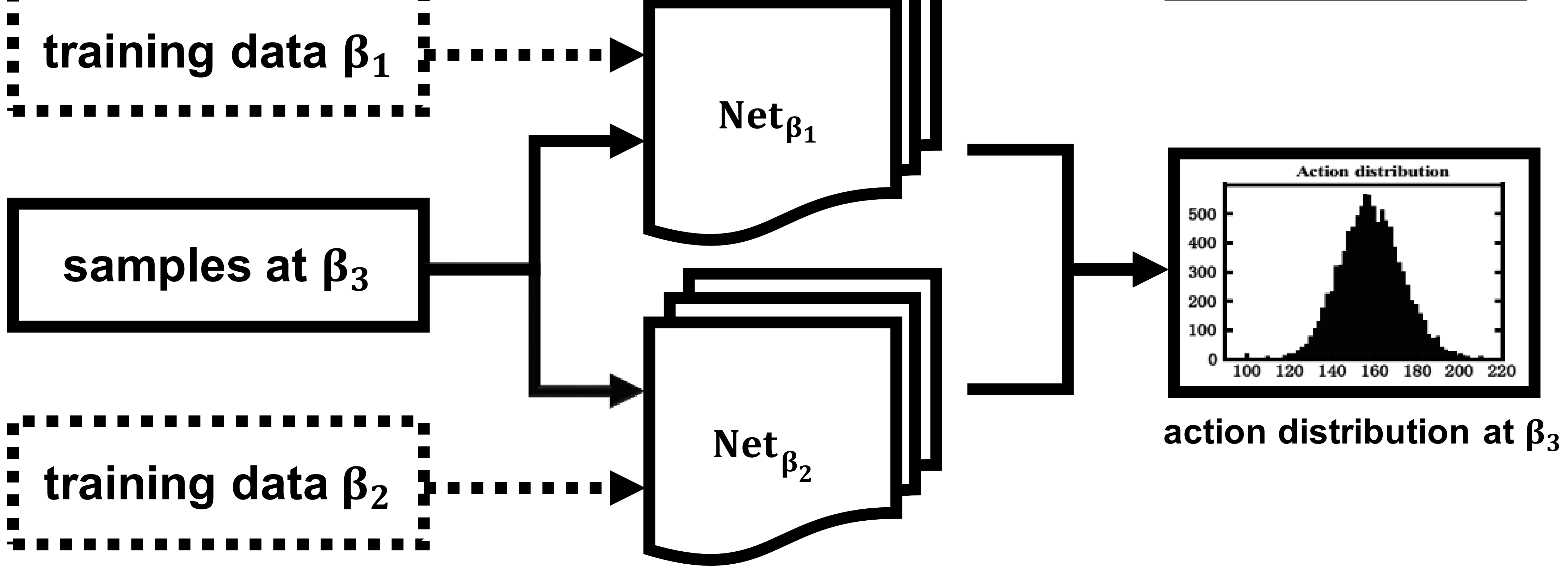
$$= \beta^{-1}K + \beta V$$

$$\Delta\tau = \beta / N_\tau$$

$$K \equiv N_\tau \sum (\Delta x)^3 (\Delta\Phi)^2$$

$$V \equiv N_\tau^{-1} \sum (\Delta x)^3 [(\nabla\Phi)^2 + V(\Phi)]$$

arXiv:2405.10493



$$S_1[\Phi] = \beta_1^{-1}K[\Phi] + \beta_1 V[\Phi] + C_1$$

$$S_2[\Phi] = \beta_2^{-1}K[\Phi] + \beta_2 V[\Phi] + C_2,$$

$$S_3[\Phi] = \frac{\beta_1(\beta_3^2 - \beta_2^2)}{\beta_3(\beta_1^2 - \beta_2^2)} S_1 + \frac{\beta_2(\beta_1^2 - \beta_3^2)}{\beta_3(\beta_1^2 - \beta_2^2)} S_2 + C_3$$

$$C_3 = \frac{\beta_1(\beta_2^2 - \beta_3^2)}{\beta_3(\beta_1^2 - \beta_2^2)} C_1 + \frac{\beta_2(\beta_3^2 - \beta_1^2)}{\beta_3(\beta_1^2 - \beta_2^2)} C_2$$

Finite-Temperature Fields

0+1 D Quantum Field

arXiv:2405.10493

$$\mathscr{L} = \frac{1}{2}\left(\frac{dx}{d\tau}\right)^2 + V_k(x) \qquad V_k(x) = \frac{\lambda_k}{4}\left(x^2 - \frac{\mu_k^2}{2k}\right)^2$$

$$Z = \int_{x(\beta)=x(0)} Dx\ e^{-S_E[x(\tau)]}$$

$$= \int \prod_{j=-N+1}^{N+1} \frac{dx_j}{\sqrt{2\pi a}} \times \exp\left\{-\sum_{i=-N+1}^{N+1}\left[\frac{(x_{i+1}-x_i)^2}{2a} + aV_k(x_i)\right]\right\}$$

$$S_E[x(\tau)] = \int_0^\beta d\tau\ \mathscr{L}_E[x(\tau)] = \int_0^\beta d\tau\ \left[\frac{1}{2}\left(\frac{dx}{d\tau}\right)^2 + V_k(x)\right]$$

$$x(\tau) = \pm\frac{\mu_k}{\sqrt{\lambda_k}}\tanh\left[\frac{\mu_k}{\sqrt{2}}(\tau-\tau_0)\right]$$

Kink/Anti-Kink solutions reach $\pm\mu_k/\sqrt{\lambda_k}$ at $\tau = \pm\infty$

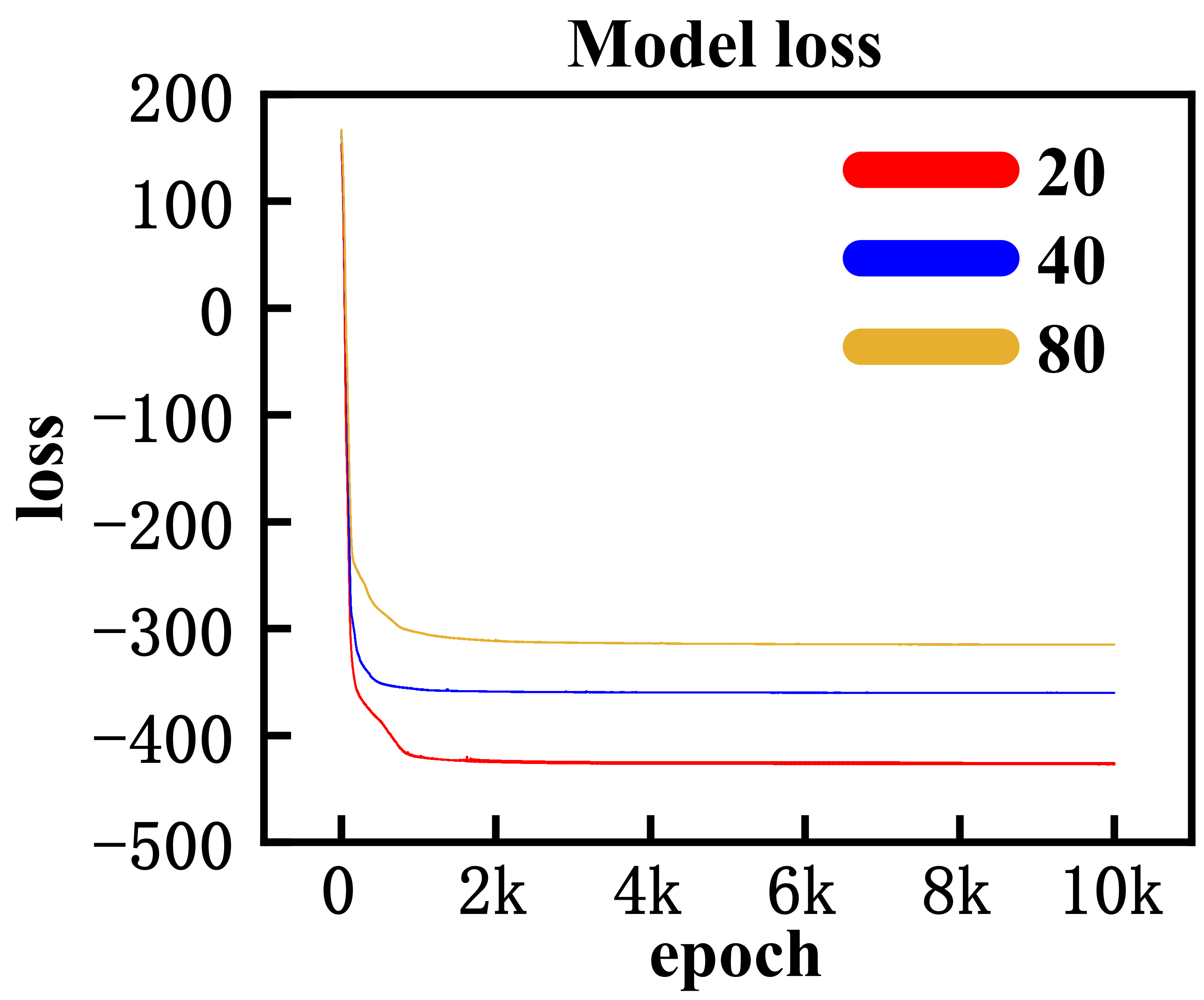
Numerical Simulations

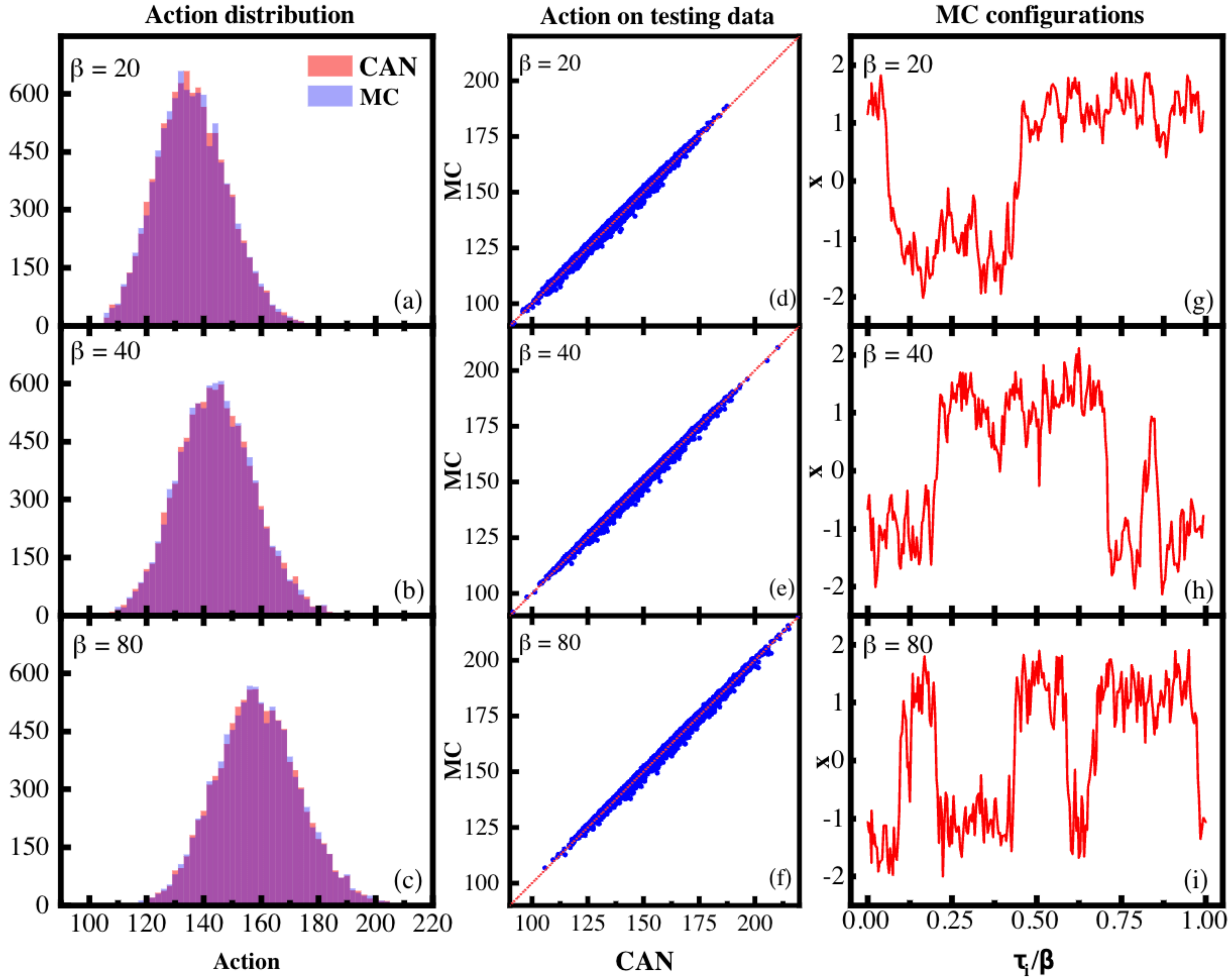
$$\lambda_k = 4 \qquad \mu_k/\sqrt{\lambda_k} = 1.4$$

$$\beta = T^{-1} = 80,40,20 \qquad N_{MC} = 5\times 10^6$$

Finite-Temperature Fields

0+1 D Quantum Field

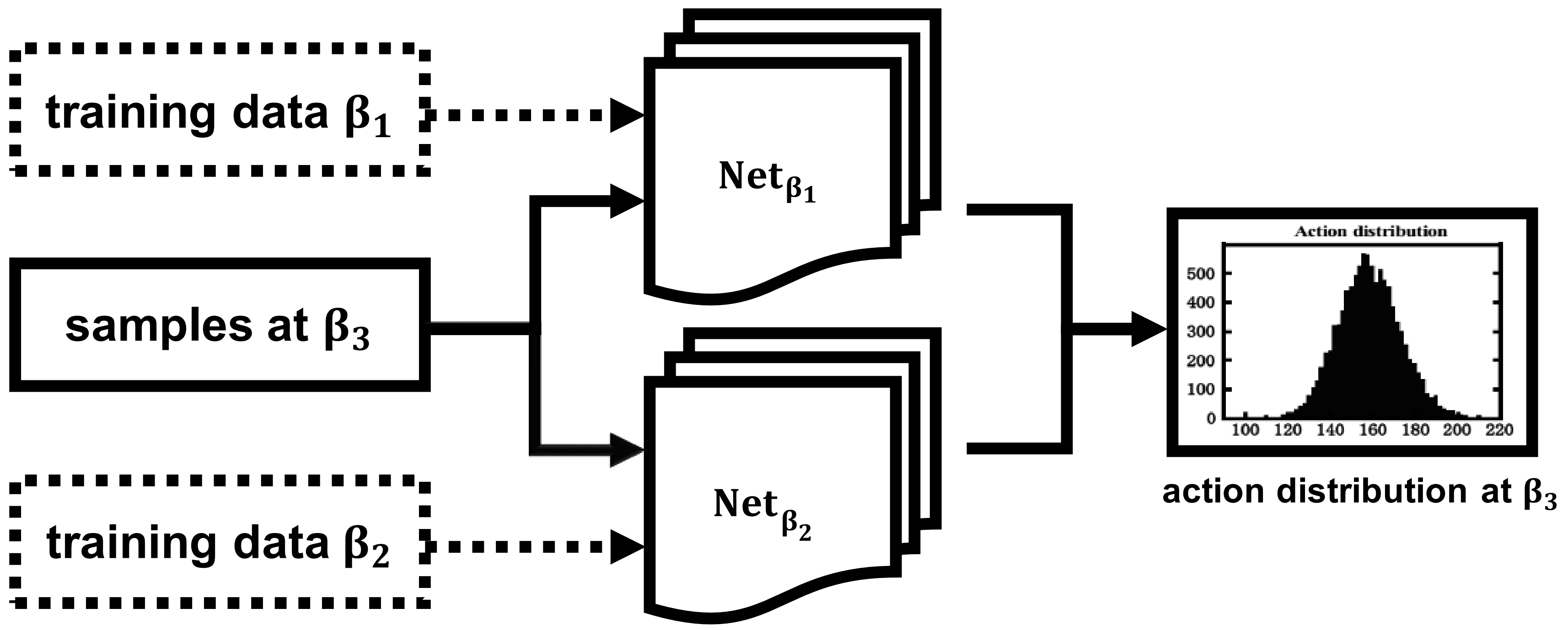


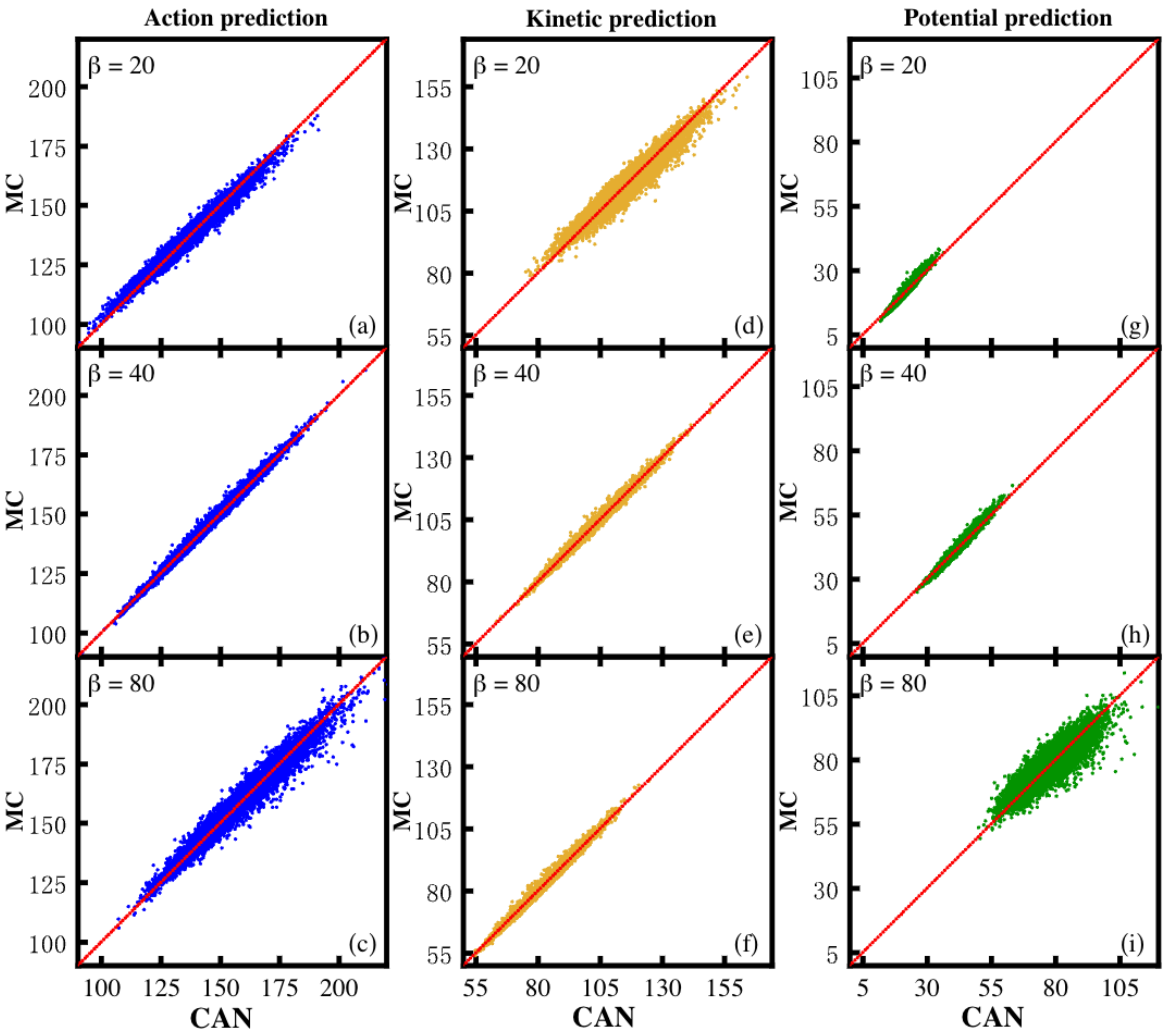


iv:2405.10493

Finite-Temperature Fields

0+1 D Quantum Field

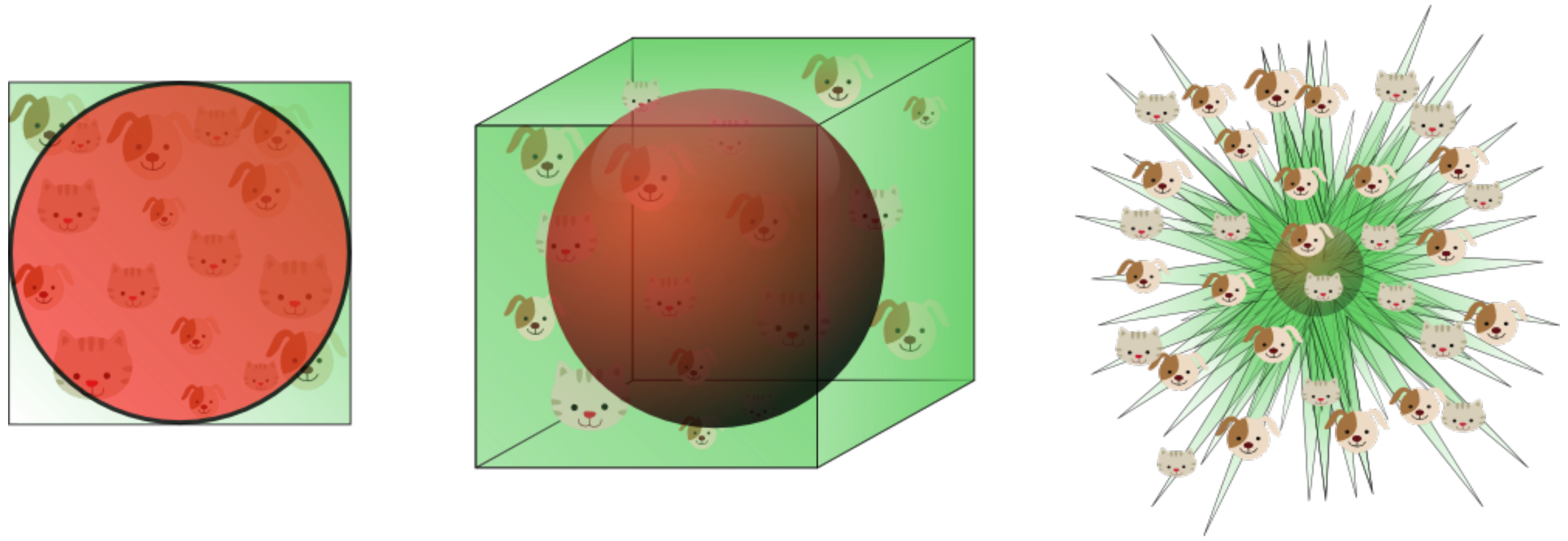




arXiv:2405.10493

Diffusion Models

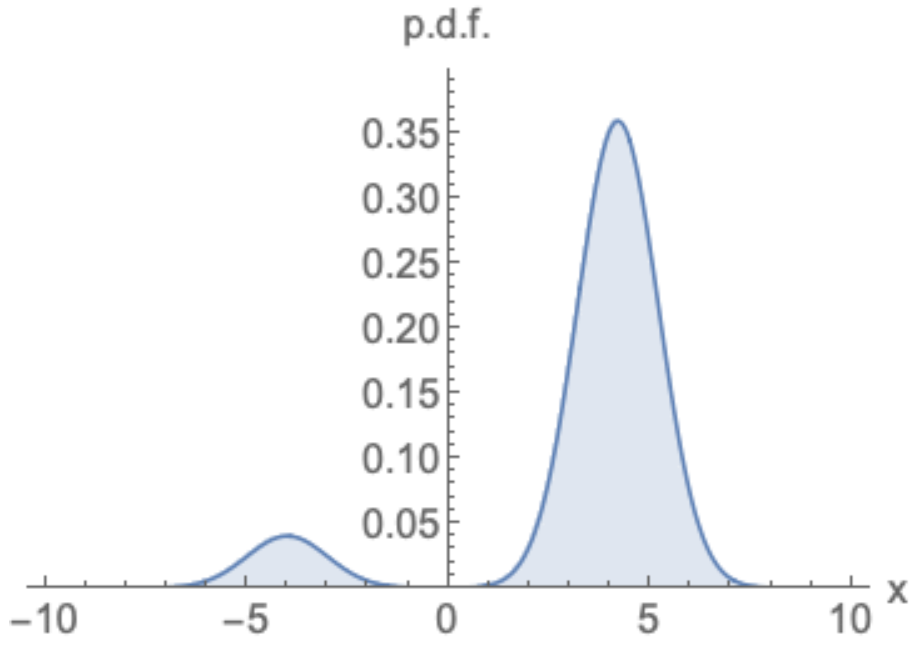
$$\mathbf{x} \sim p_{data}(\mathbf{x})$$



Curse of Dimensionality

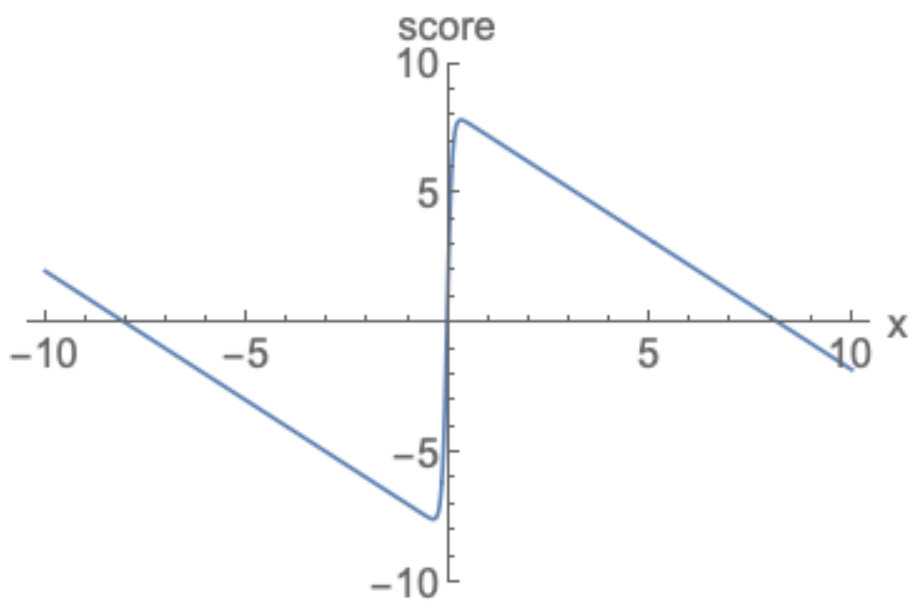
$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$$

from a **statistical phsyics** perspective



$$\nabla_\mathbf{x} \log p(\mathbf{x})$$

$$\mathbf{s}_\theta(\mathbf{x}) \rightarrow \nabla_\mathbf{x} \log p(\mathbf{x})$$



Approaching **Score Function**

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x})$$

$$= -\nabla_\mathbf{x} f_\theta(\mathbf{x}) - \underbrace{\nabla_\mathbf{x} \log Z_\theta}_{=0} = -\nabla_\mathbf{x} f_\theta(\mathbf{x})$$

Diffusion Models

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

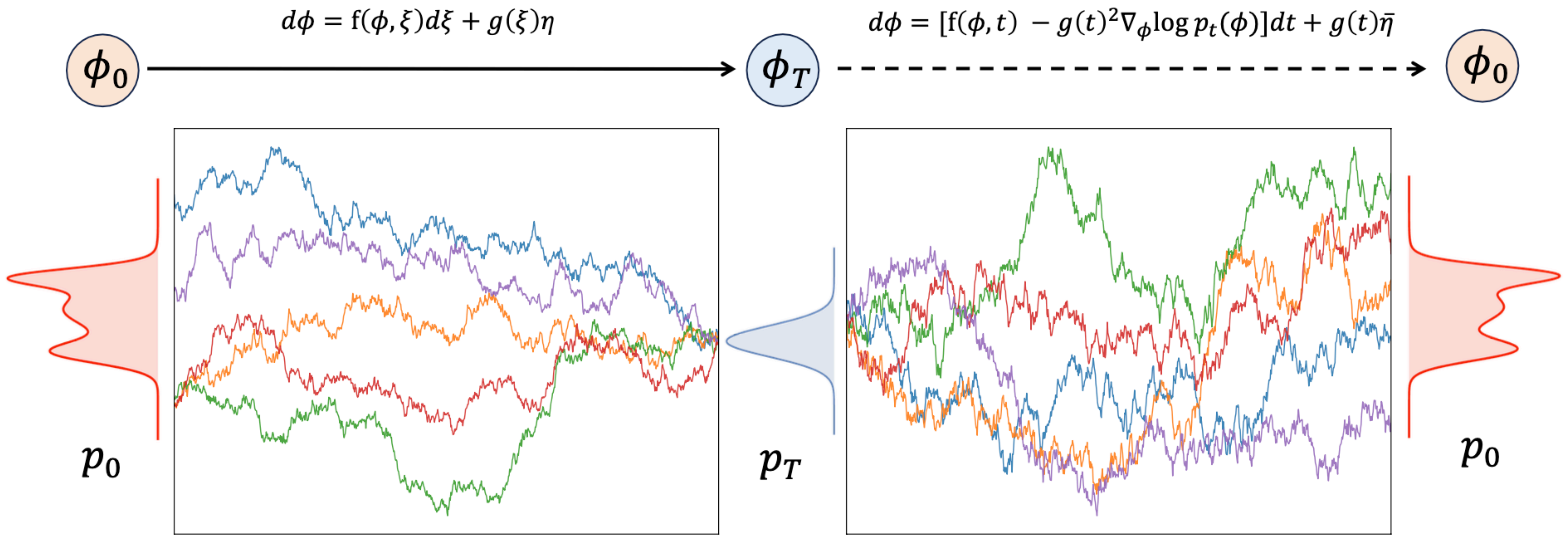
Anderson, in Stochastic Processes and their Applications, 1982

- **Forward Diffusion SDE**
  - **Drift term**: pulls towards mode
  - **Diffusion term**: injects noise

$$\frac{d\phi}{d\xi} = f(\phi, \xi) + g(\xi)\eta(\xi)$$

- **Reverse Generative Diffusion SDE**
  - Drift term is adjusted with a "**Score Function**"
  - Represent the score function with **neural networks**

$$\frac{d\phi}{dt} = \left[f(\phi, t) - g^2(t)\nabla_\phi \log p_t(\phi)\right] + g(t)\bar{\eta}(t)$$



Diffusion Models

Stochastic Quantization

Parisi G. and Wu Y. S., Sci. China, A **24**, ASITP-80-004 (1980).

$$\frac{\partial\phi(x,\tau)}{\partial\tau} = -\frac{\delta S_E[\phi]}{\delta\phi(x,\tau)} + \eta(x,\tau)$$

$$\langle\eta(x,\tau)\rangle = 0, \quad \langle\eta(x,\tau)\eta(x',\tau')\rangle = 2\alpha\delta(x-x')\delta(\tau-\tau')$$

τ: fictitious time, α: diffusion constant

- **Fokker-Planck equation**

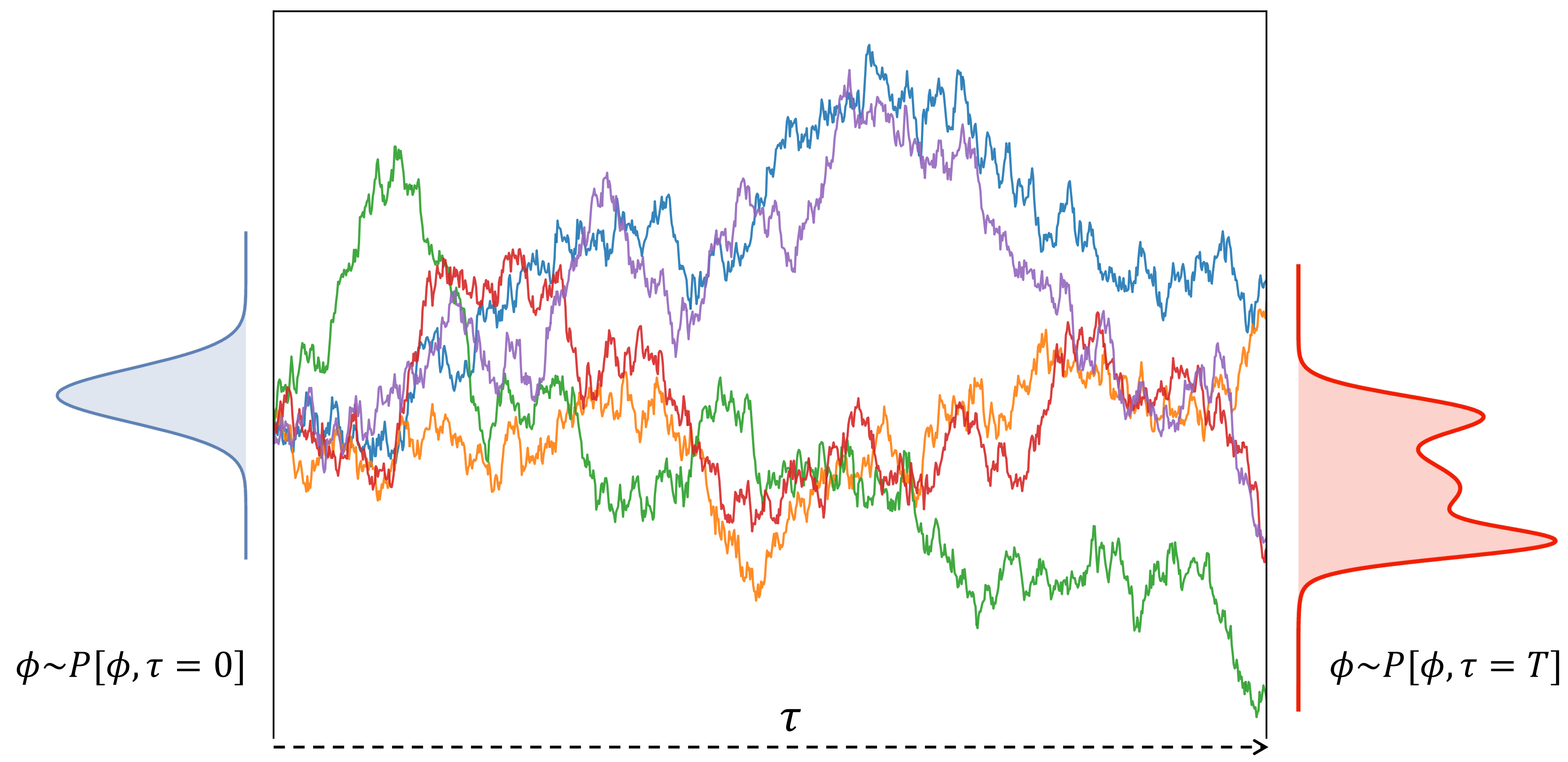
$$\frac{\partial P[\phi,\tau]}{\partial\tau} = \alpha\int d^n x \left\{\frac{\delta}{\delta\phi}\left(\frac{\delta}{\delta\phi} + \frac{\delta S_E}{\delta\phi}\right)\right\} P[\phi,\tau]$$

Equilibrium solution (long-time limit),

$$P_{\text{eq}}[\phi] \propto e^{-\frac{1}{\alpha}S_E[\phi]}$$

- Set the diffusion constant as $\alpha = \hbar$

$$P_{\text{eq}}[\phi] \sim e^{-\frac{1}{\hbar}S_E[\phi]} = P_{\text{quantum}}[\phi]$$



Thermal equilibrium limit → Quantum distribution

1. No need gauge-fixing!
2. Can handle fermionic fields naturally
→(Complex Langevin method)
....

P. H. Damgaard and H. Hüffel, Stochastic Quantization, Phys. Rept. 152, 227 (1987).
M. Namiki, Basic Ideas of Stochastic Quantization, PTPS 111, 1 (1993).
G. Aarts, L. Bongiovanni, E. Seiler, D. Sexty, and I.-O. Stamatescu, Eur. Phys. J. A **49**, 89 (2013).

Diffusion Models

DMs as SQ

JHEP 05(2024)060

- **Diffusion models(Reverse SDE):**

$$\frac{d\phi}{dt} = -g(t)^2 \nabla_\phi \log p_t(\phi) + g(t)\bar{\eta}$$

- **Define:** $\tau \equiv T - t (d\tau \equiv -dt)$

$$\frac{d\phi}{d\tau} = g_\tau^2 \nabla_\phi \log q_\tau(\phi) + g_\tau \bar{\eta}$$

$$\phi(\tau_{n+1}) = \phi(\tau_n) + g_\tau^2 \nabla_\phi \log q_{\tau_n}[\phi(\tau_n)]\Delta\tau + g_\tau \sqrt{\Delta\tau}\bar{\eta}(\tau_n)$$

introducing **Noise scale:** $\langle \bar{\eta}^2 \rangle \equiv 2\bar{\alpha}$, **time scale:** $g_\tau^2 \Delta\tau$

- **FP equation**

$$\frac{\partial p_\tau(\phi)}{\partial \tau} = \int d^n x \left\{ g_\tau^2 \bar{\alpha} \frac{\delta}{\delta\phi} \left(\frac{\delta}{\delta\phi} + \frac{1}{\bar{\alpha}} \nabla_\phi S_{\mathrm{DM}} \right) \right\} p_\tau(\phi)$$

$$\nabla_\phi S_{\mathrm{DM}} \equiv -\nabla_\phi \log q_\tau(\phi)$$

$$p_{eq}(\phi) \propto e^{-\frac{S_{DM}}{\bar{\alpha}}}$$

$$p_{\tau=T}(\phi) \to P[\phi, T]$$

$$O(\bar{\alpha}) \sim O(\hbar)$$

The reverse mode of
a well-trained diffusion model at $\tau \to T$ serves as
the stochastic quantization for the input

Diffusion Models

DM for Scalar Field

JHEP 05(2024)060

- **Euclidean action** on lattice

$$S_E = \sum_x a^d [\sum_{\mu=1}^{d} \frac{(\phi_0(x+a\hat{\mu}) - \phi_0(x))^2}{a^2} + \frac{m_0^2}{2}\phi_0^2 + \frac{\lambda_0}{4!}\phi_0^4]$$

- Dimensionless form

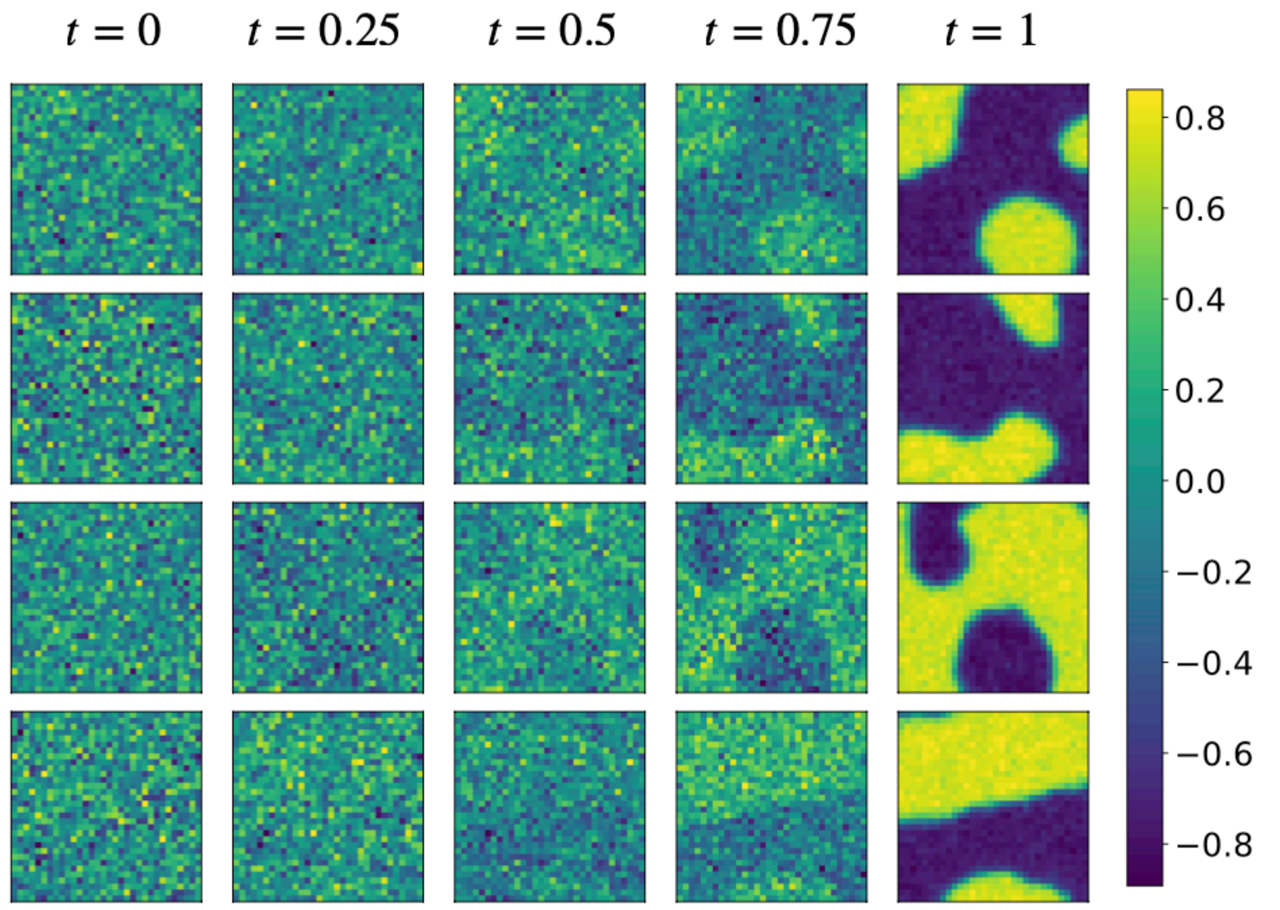
$$S_E = \sum_x [-2\kappa \sum_{\mu=1}^{d} \phi(x)\phi(x+\hat{\mu}) + (1-2\lambda)\phi(x)^2 + \lambda\phi(x)^4]$$

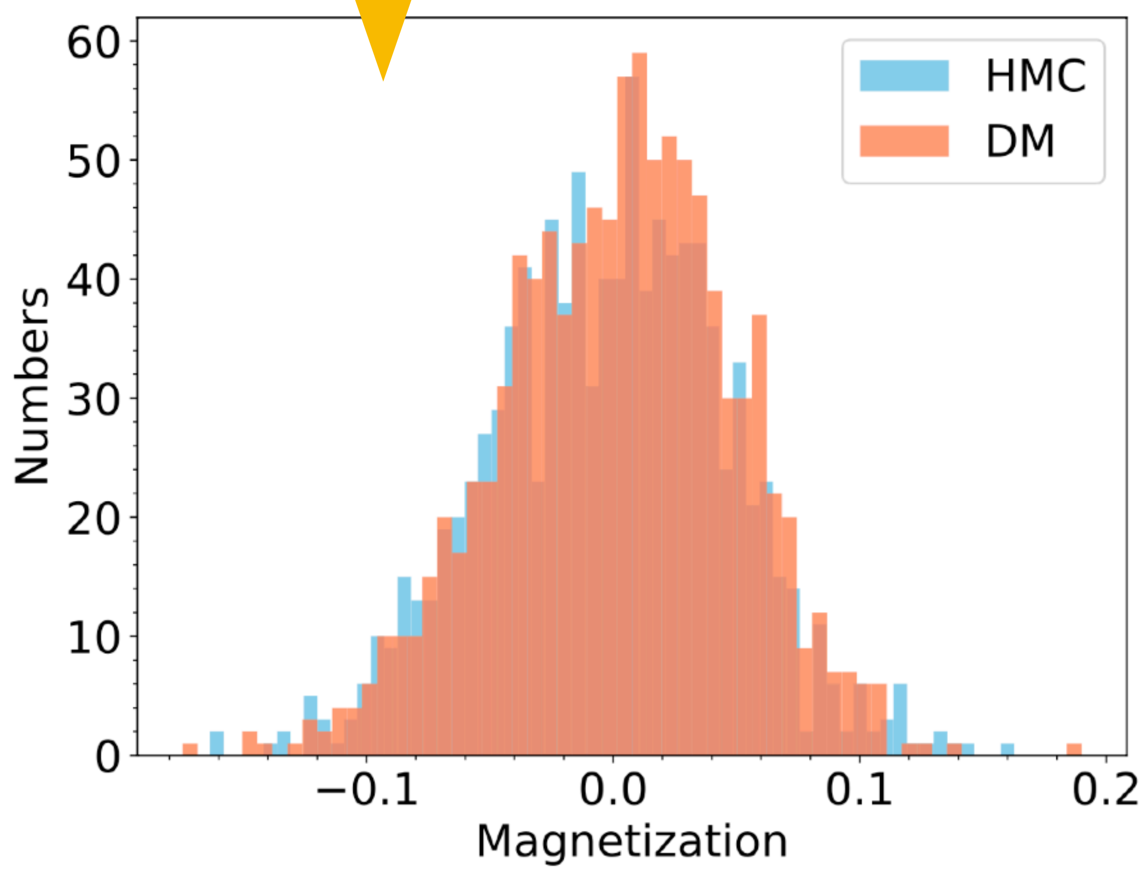
$$a^{\frac{d-2}{2}}\phi_0 = (2\kappa)^{1/2}\phi$$

$$(am_0)^2 = \frac{1-2\lambda}{\kappa} - 2d, \quad a^{-d+4}\lambda_0 = \frac{6\lambda}{\kappa^2}$$

- Hopping parameter κ, Coupling constant λ

- **Diffusion models**
  - $T = 1.0, \sigma = 25$
- **Data generation**
  - 2-d 32×32 lattice; Hamiltonian Monte Carlo(HMC); 5120 configurations for training.
  - Broken phase: $\kappa = 1.0, \lambda = 0.022$
  - Symmetric phase: $\kappa = 0.21, \lambda = 0.022$





data-set	$\langle M \rangle$	χ_2	U_L
Training(HMC)	0.0012± 0.0007	2.5160 ± 0.0457	0.1042 ± 0.0367
Testing(HMC)	0.0018 ± 0.0015	2.4463 ± 0.1099	-0.0198 ± 0.1035
Generated(DM)	0.0017± 0.0015	2.4227 ± 0.1035	0.0484 ± 0.0959

Physics-Conditioned Diffusion Models

NeurIPS2024 ML4PS Workshop **with Qianteng Zhu, Gert Aarts, etc.**

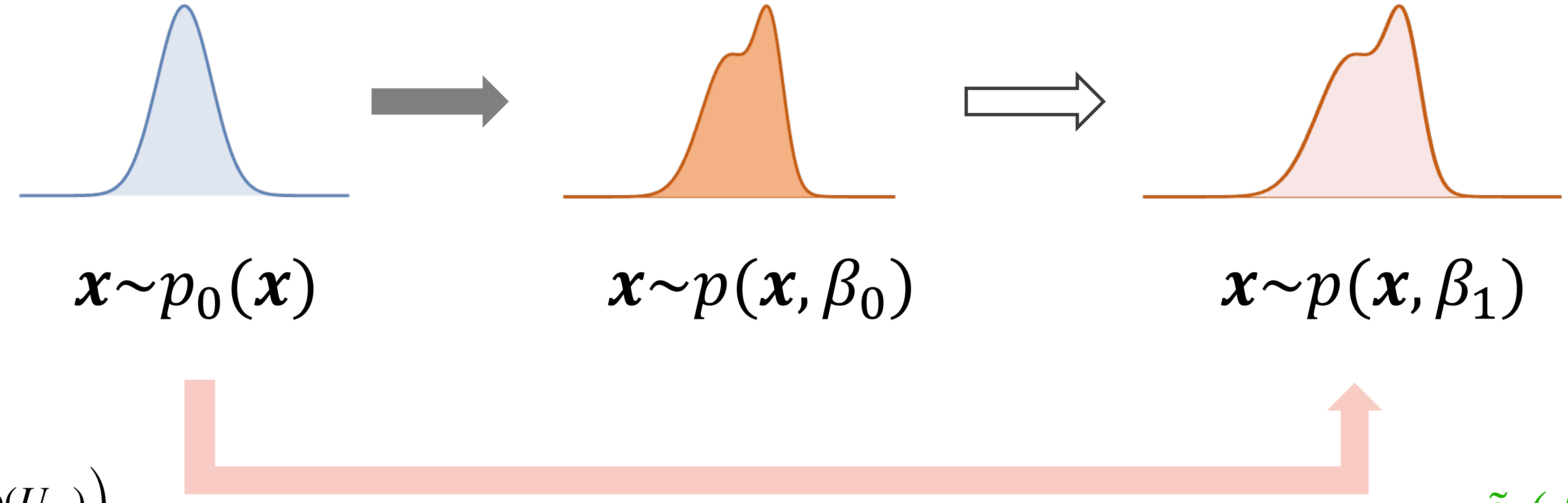
$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)} + \eta(x,\tau)$$

Drift Term

$$\frac{d\phi}{dt} = -\sigma^{2t}\mathbf{s}_{\hat{\theta}}(\phi, t) + \sigma^t \bar{\eta}(t)$$

Score Fucntion

$$-\frac{\delta S}{\delta \boldsymbol{x}}$$



$$\boldsymbol{x} \sim p_0(\boldsymbol{x}) \qquad \boldsymbol{x} \sim p(\boldsymbol{x}, \beta_0) \qquad \boldsymbol{x} \sim p(\boldsymbol{x}, \beta_1)$$

e.g.,

$$S = \beta \sum_{\square} \left(1 - \mathbf{Re}(U_{\square})\right)$$

$$-\frac{\beta_1}{\beta_0}\frac{\delta S}{\delta \boldsymbol{x}}$$

$$\tilde{\mathbf{s}}_{\hat{\theta}}(\phi, t) \equiv \beta \mathbf{s}_{\hat{\theta}}(\phi, t)$$

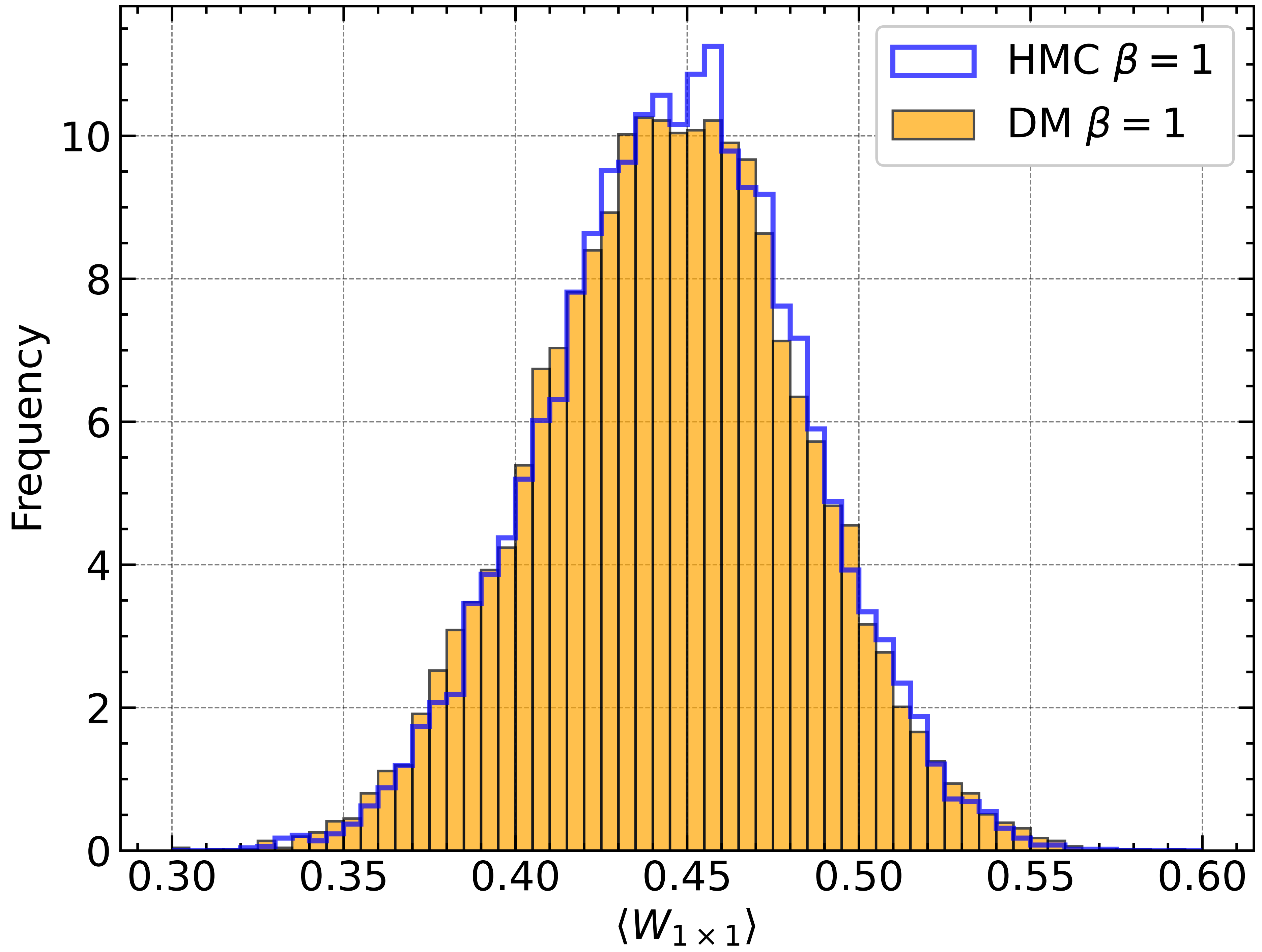
Gauge Fields

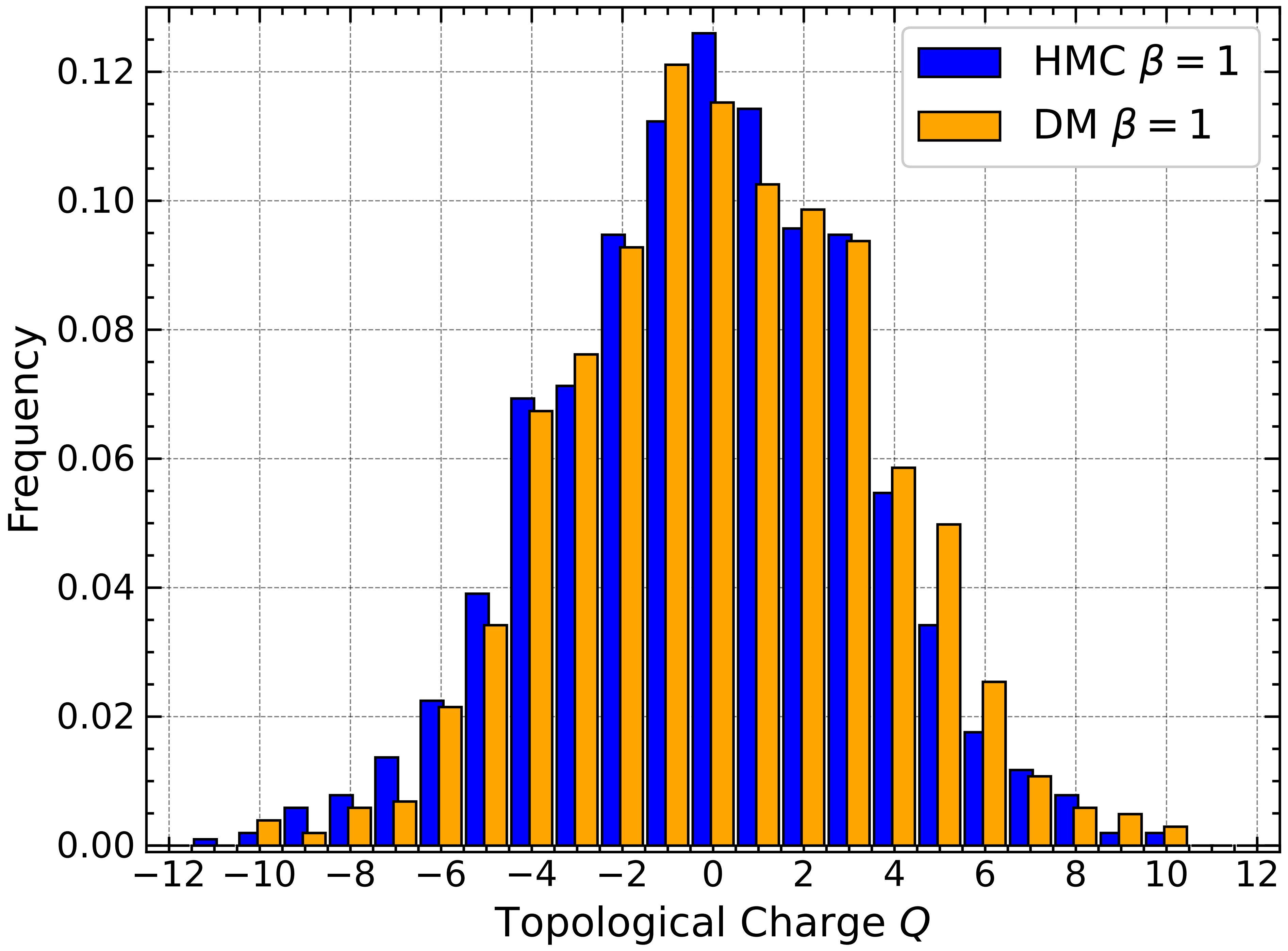
2D U(1) Gauge Field

$$S = \beta \sum_{\square} \left(1 - \mathbf{Re}(U_{\square})\right)$$

plaquette

$$U_{\square} = U_{x,\mu} U_{x+\hat{\mu},\nu} U^{\dagger}_{x+\hat{\nu},\mu} U^{\dagger}_{x,\nu}$$





$$W(C) = \mathbf{Tr}\left(\prod_{(x,\mu)\in C} U_{x,\mu}\right)$$

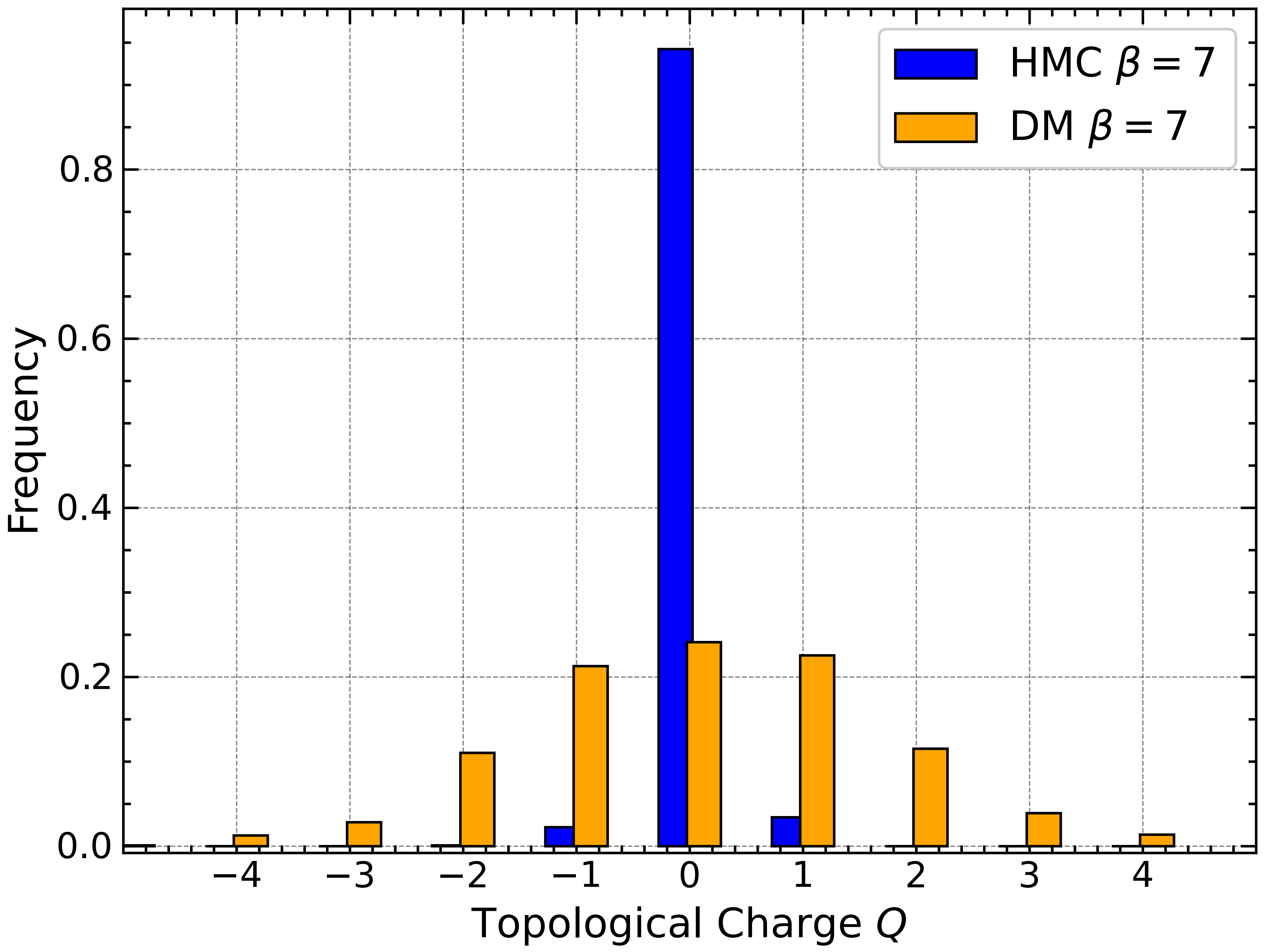
Learned at $\beta = 1$ with **10,240** cofigurations, L = 16
Generated 1024 configs for testing

$$Q = \frac{1}{2\pi} \sum_{x} F_{01}(x)$$

Gauge Fields

2D U(1) Gauge Field

Topological Freezing?



Generated at $\beta = 7$ with **1024** cofigurations, L = 16

Eur. Phys. J. C (2021) 81:873

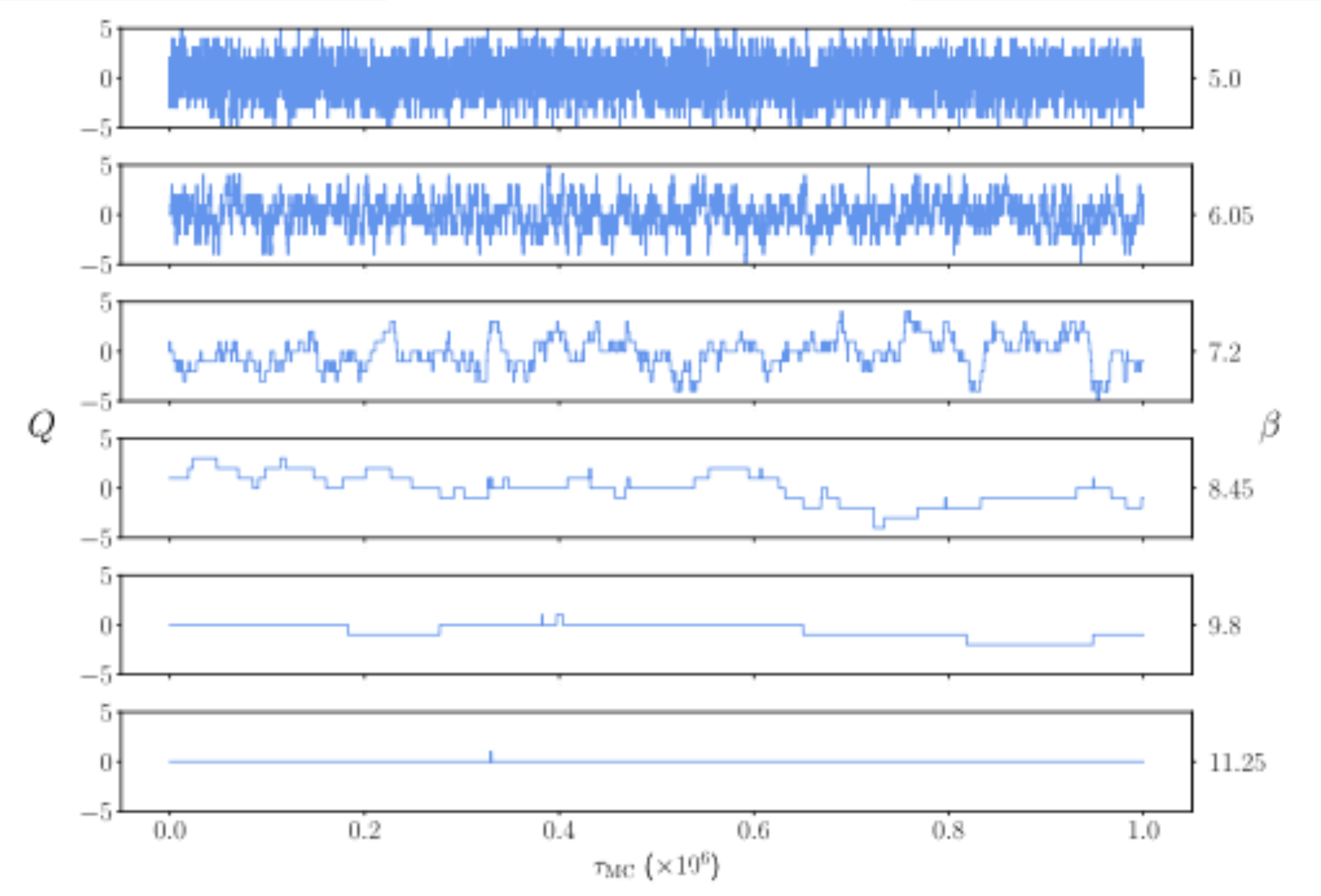
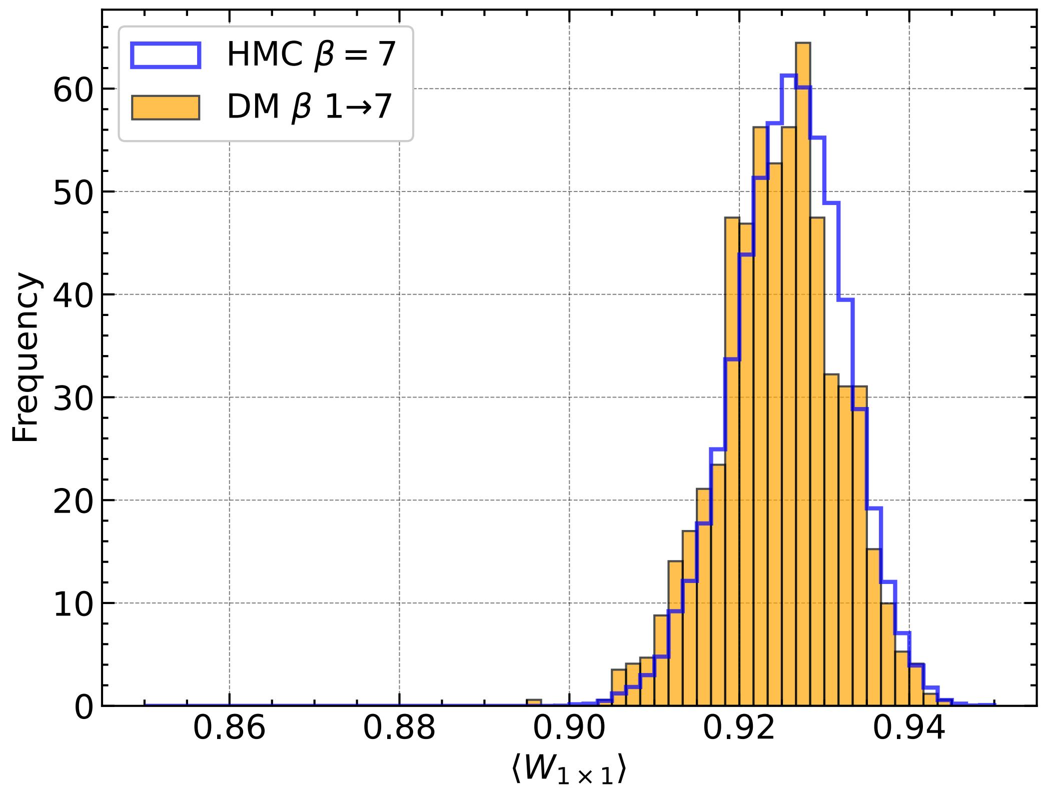
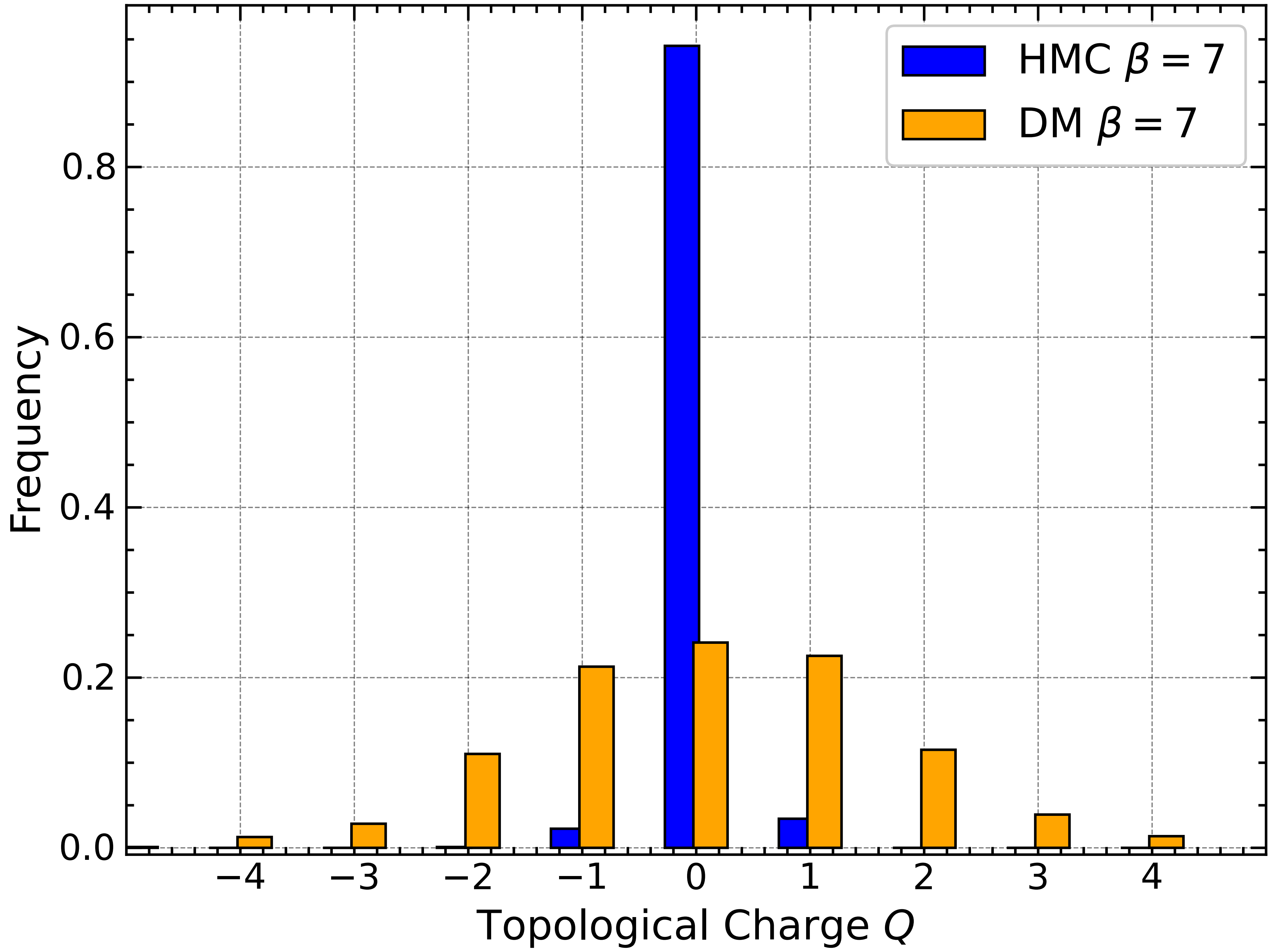


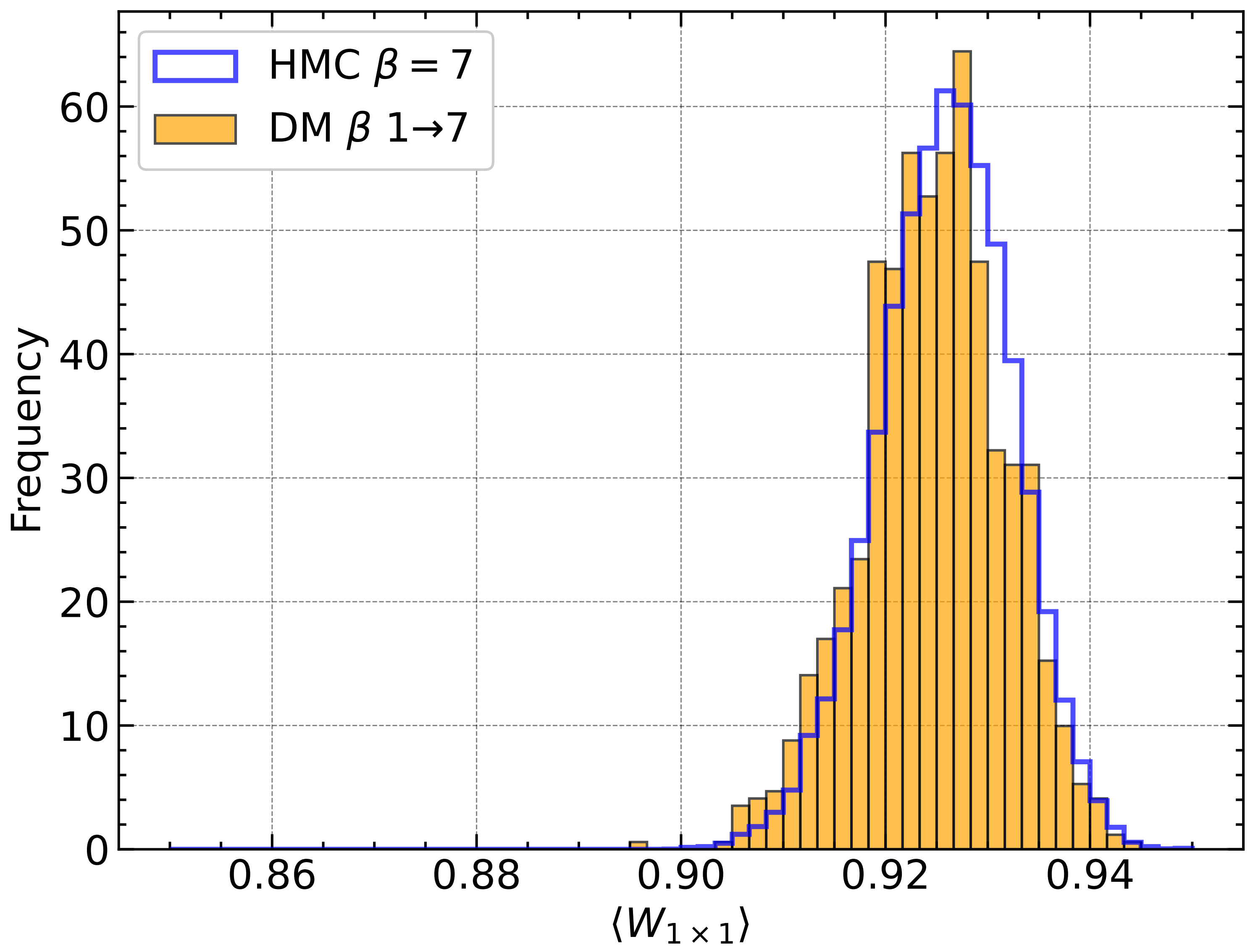
Fig. 1 Monte Carlo history of the topological charge Q for increasing values of β in a Markov chain of 10^6 HMC configurations



Gauge Fields

2D U(1) Gauge Field





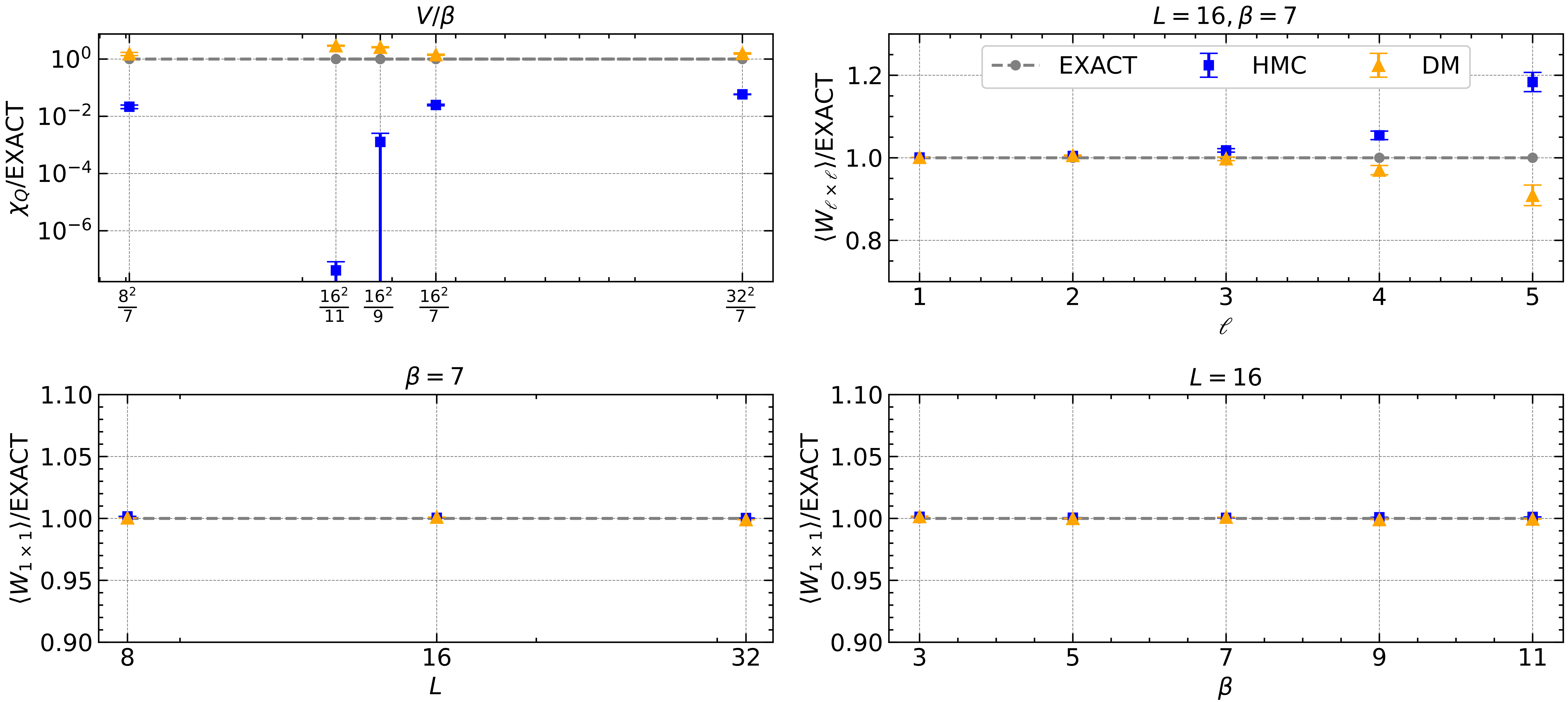
Physics-Conditioned Diffusion Models

Training at one,
Transfer and Generate at all

Generated at $\beta = 7$ with **1024** cofigurations, L = 16

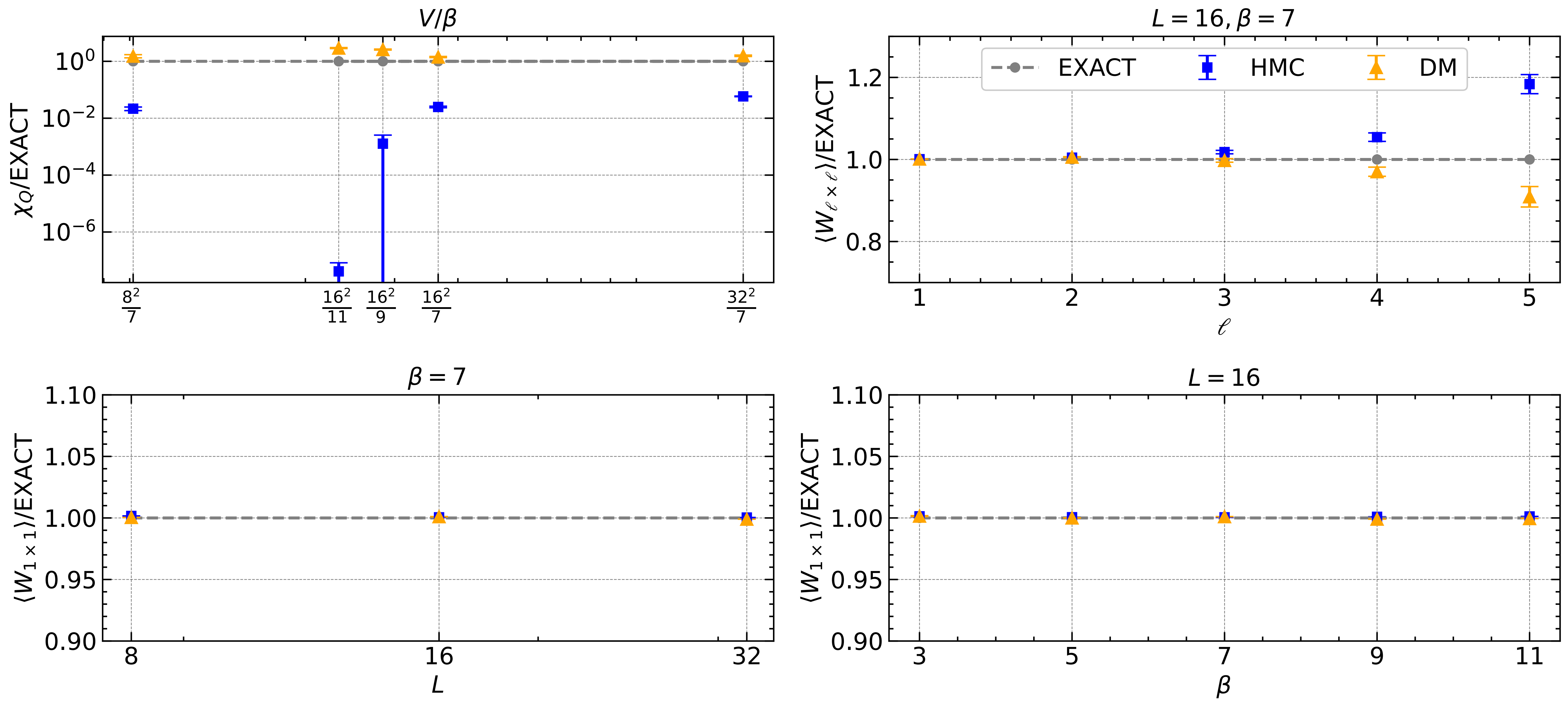
Gauge Fields

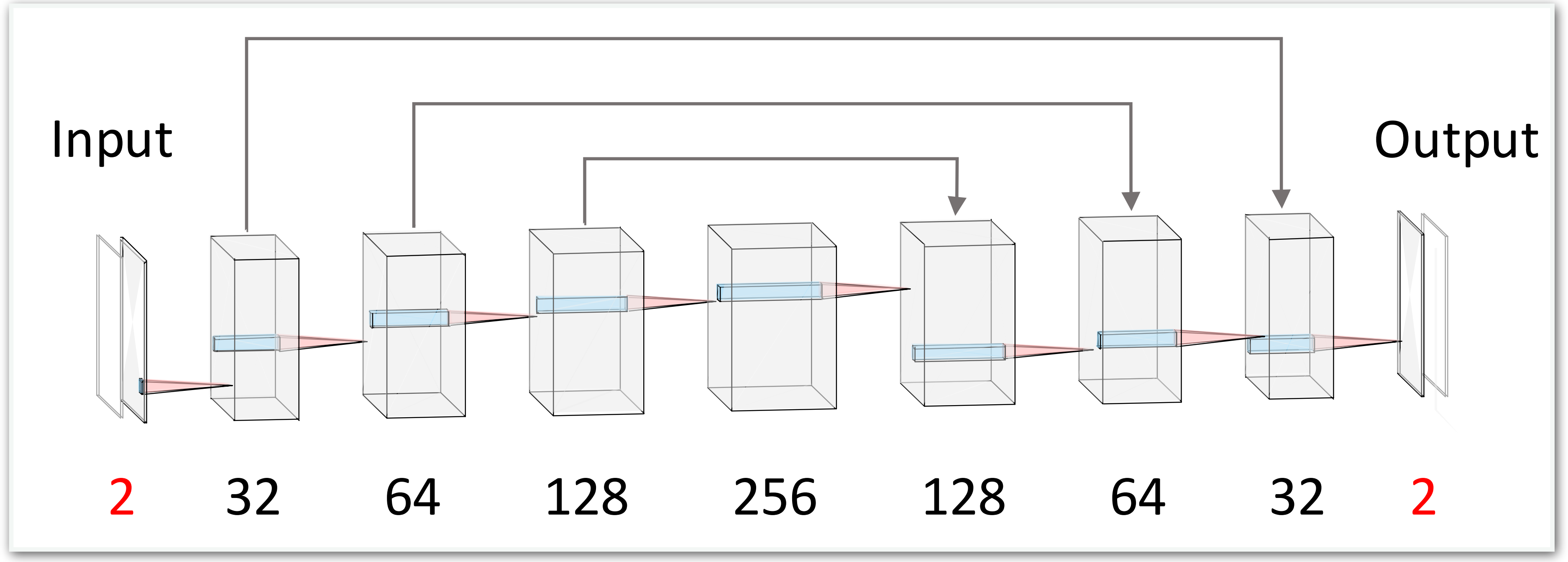
2D U(1) Gauge Field



Gauge Fields

2D U(1) Gauge Field





$$\tilde{\mathbf{s}}_{\hat{\theta}}(\phi, t) \equiv \beta \mathbf{s}_{\hat{\theta}}(\phi, t)$$

$$\frac{\partial \phi(x,\tau)}{\partial \tau} = -\frac{\delta S_E[\phi]}{\delta \phi(x,\tau)} + \eta(x,\tau)$$

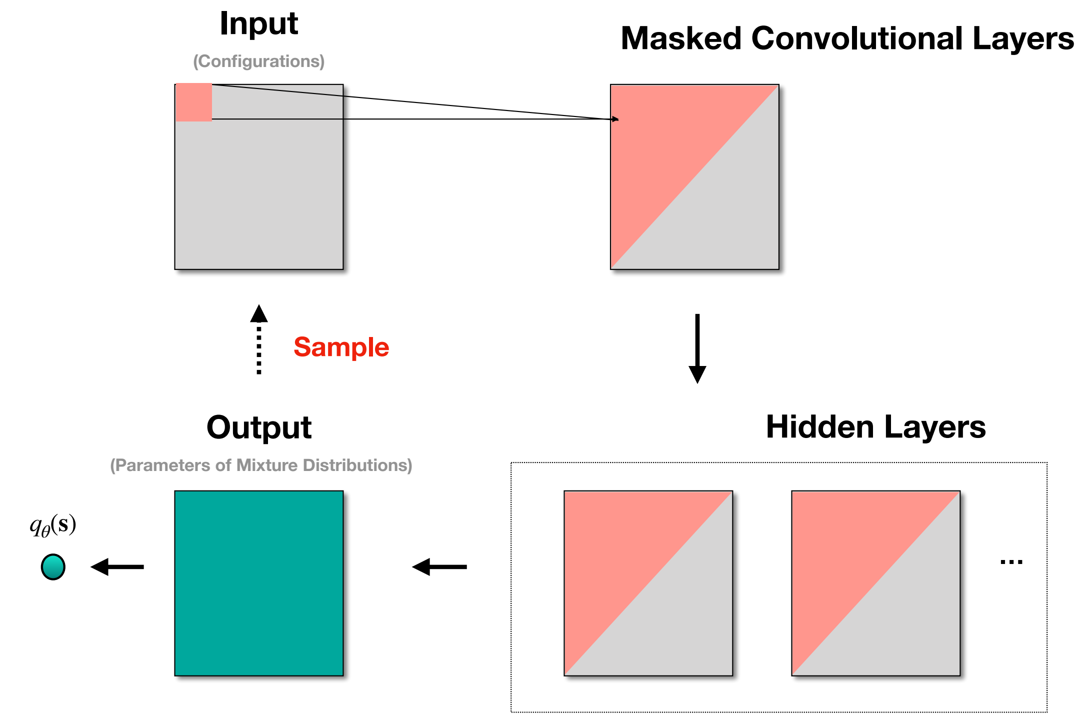
Drift Term

Different Lattice Sizes, Inverse Coupling Constants

Training at one,
Transfer and Generate at all

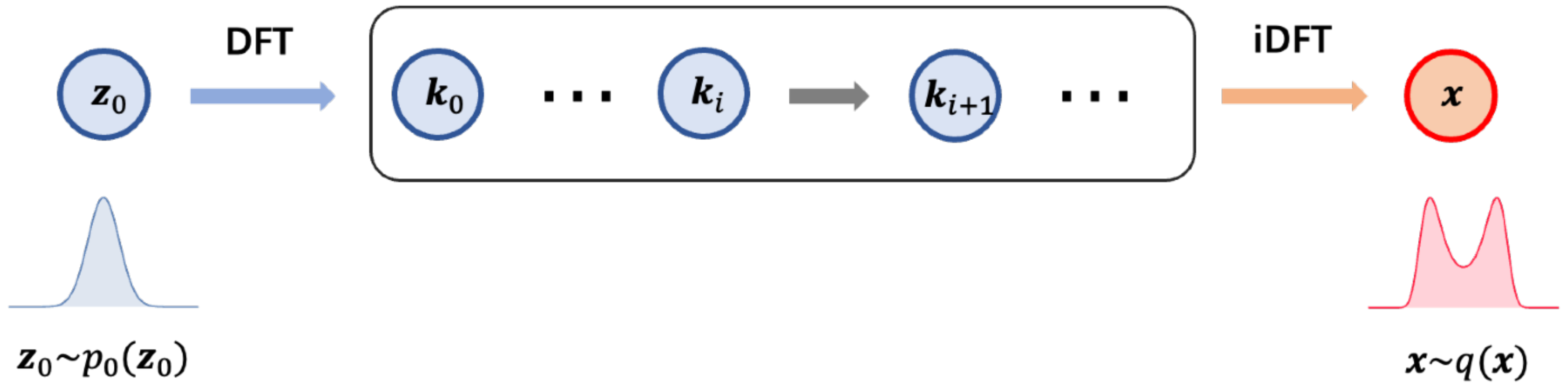
Summary II

- **Generative Models**
  - Probabilistic Models
  - Learn to Sample
  - Learn to Detect Phase Transition
  - Learn for Generating All
- **Future works**
  - 2D SU(2) **Gauge Field**
  - Complex Langevin Method(CLM)
  - DM for full QCD

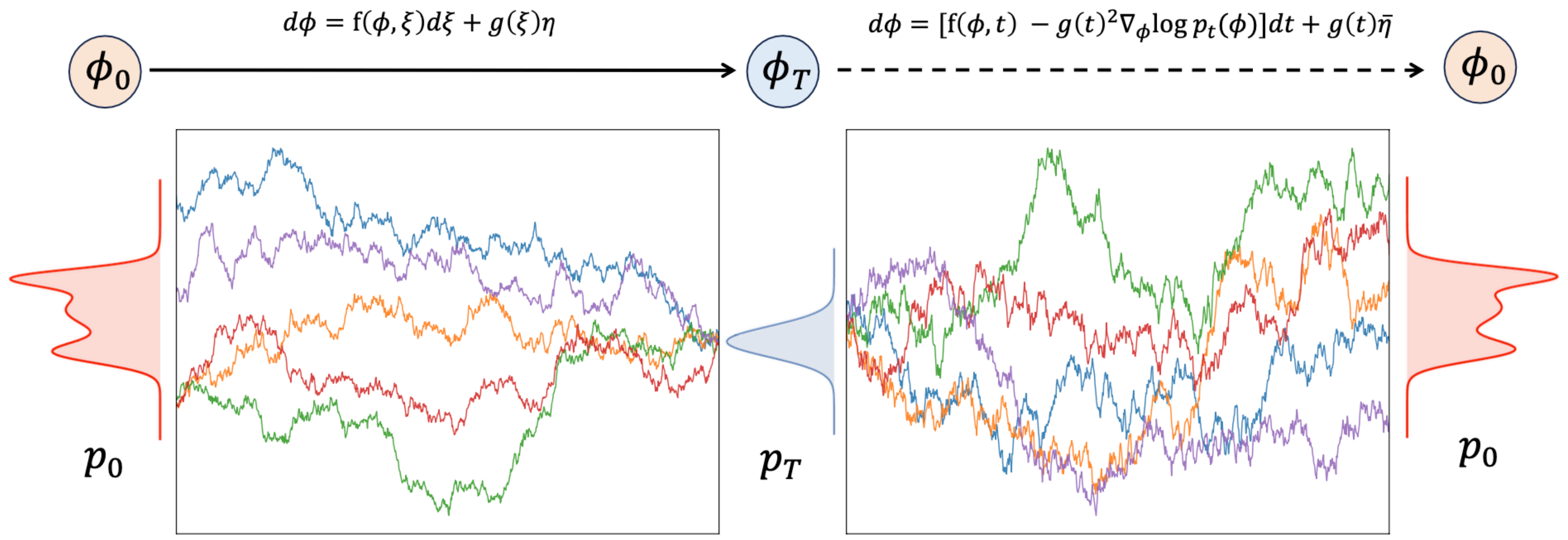


Chinese Phys. Lett. 39, 120502 (2022)

Phys. Rev. D 107, 056001



JHEP 05(2024)060



Thank You !

ML meets Physics, Opportunities and Challenges

Kouzou Sakai @Quanta Magazine

Representation Learning

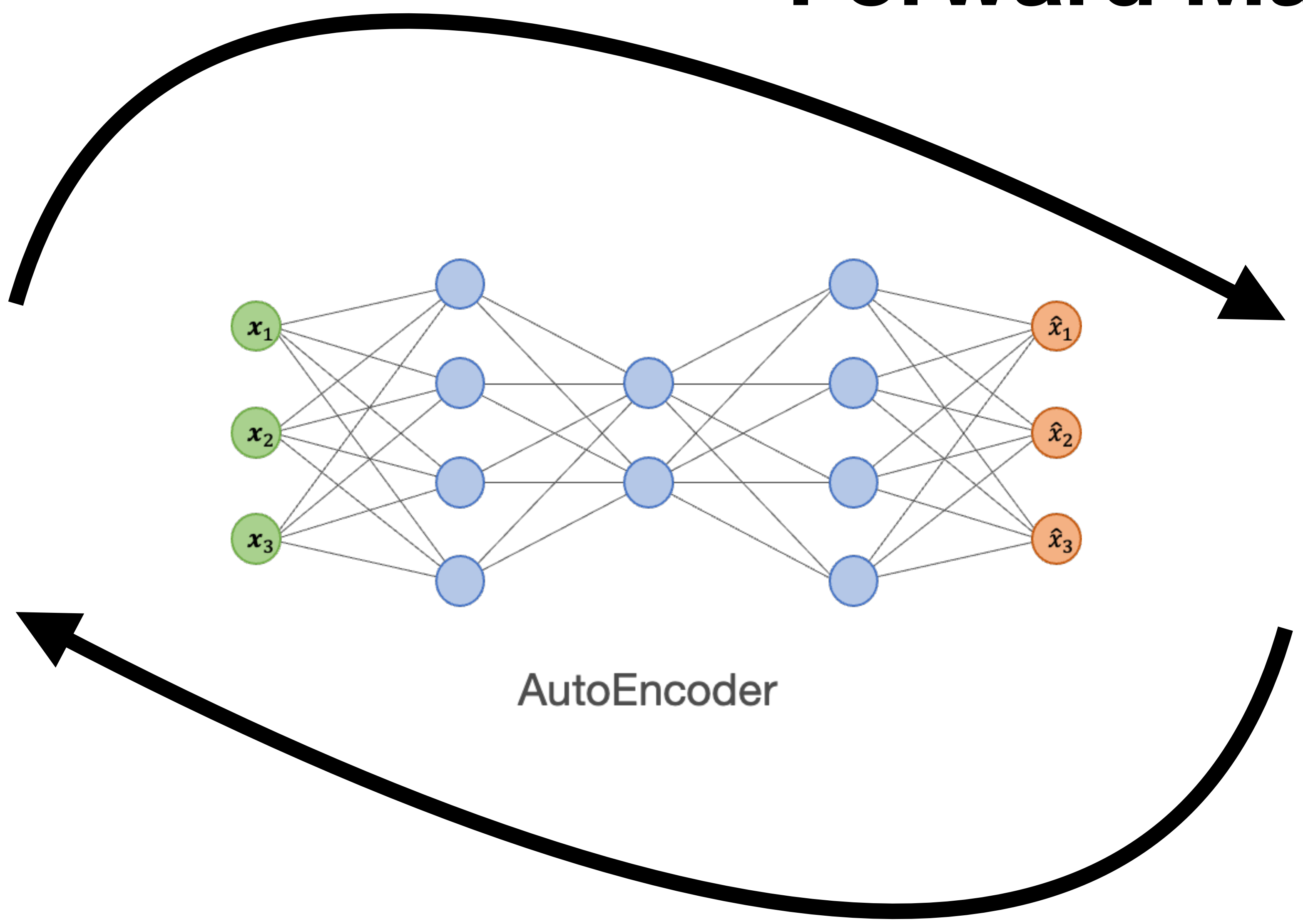
$$g_\theta : X \to Y$$

$$f_\theta : Y \to X$$

Forward Mapping, f_θ

Physics

Model Parameters/ Properties/States



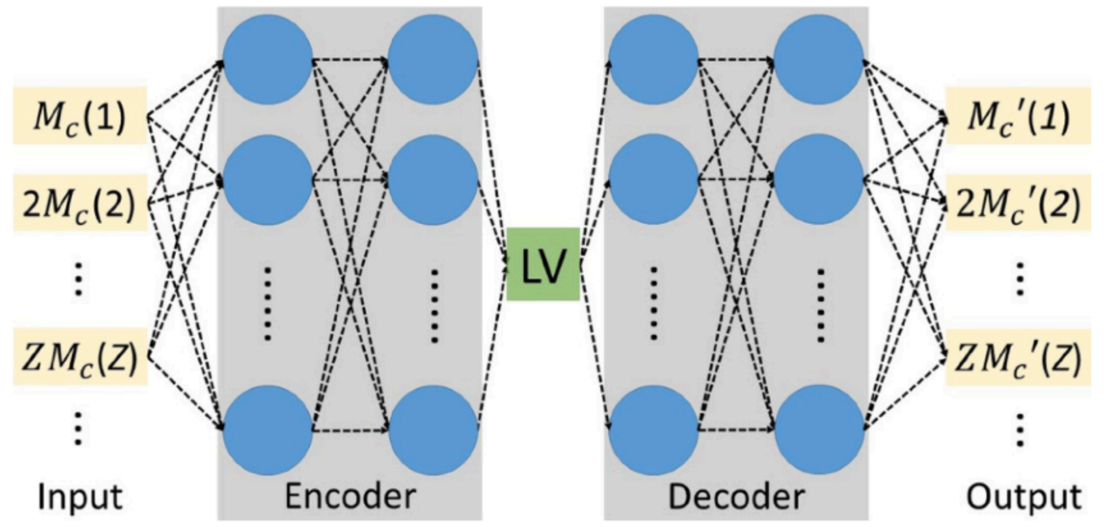
Data

Observations

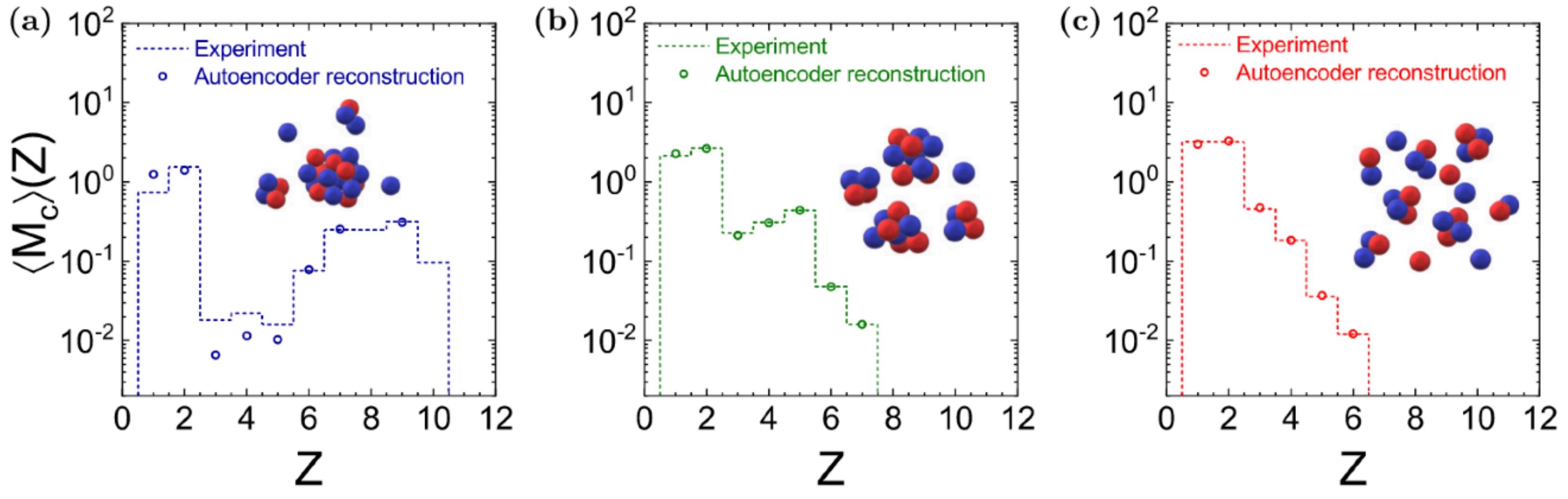
Inverse Mapping, g_θ

Representation Learning

Recognizing nuclear liquid-gas phase transition

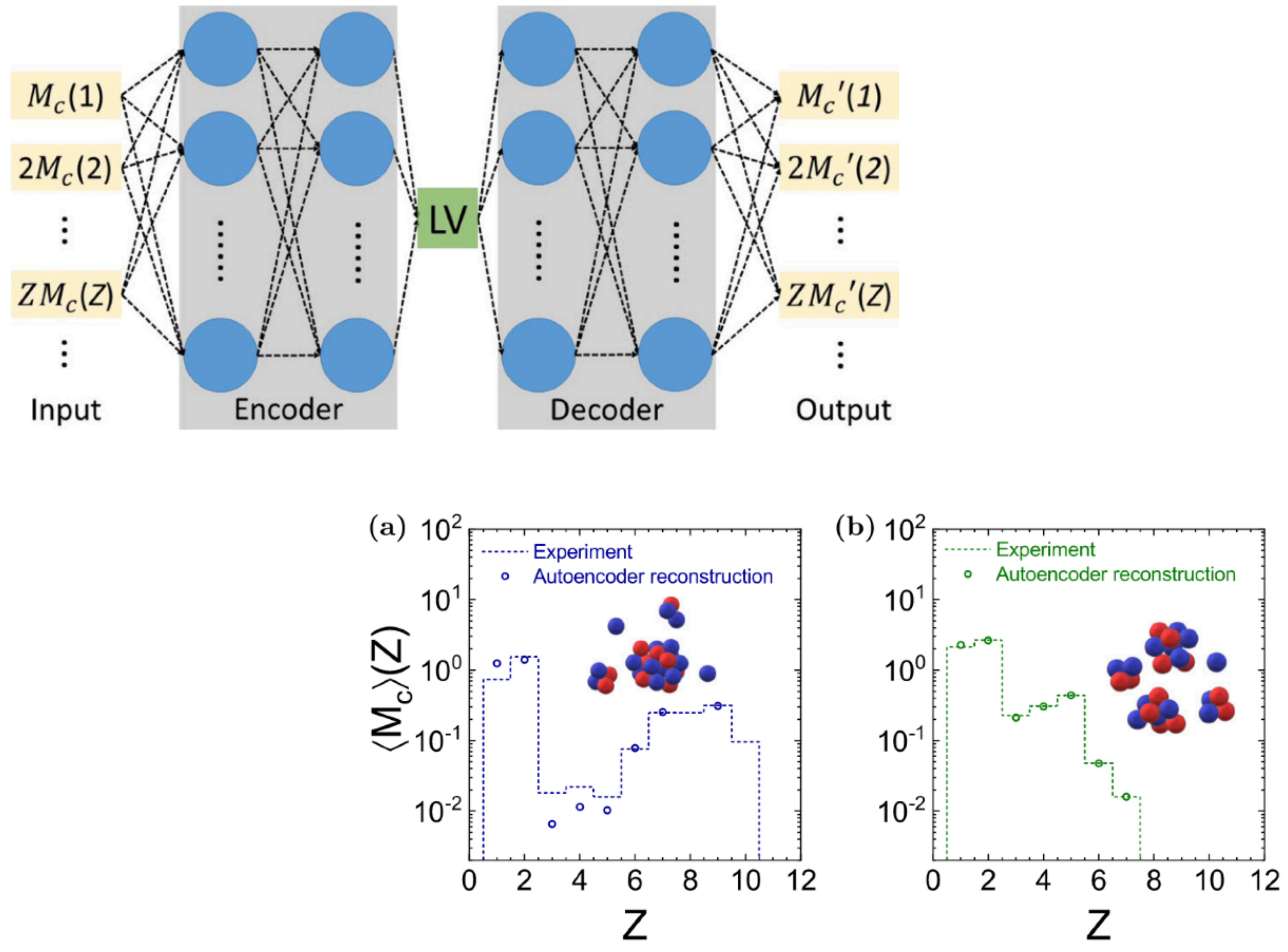


Anomaly Detection
Latent Variable Extraction



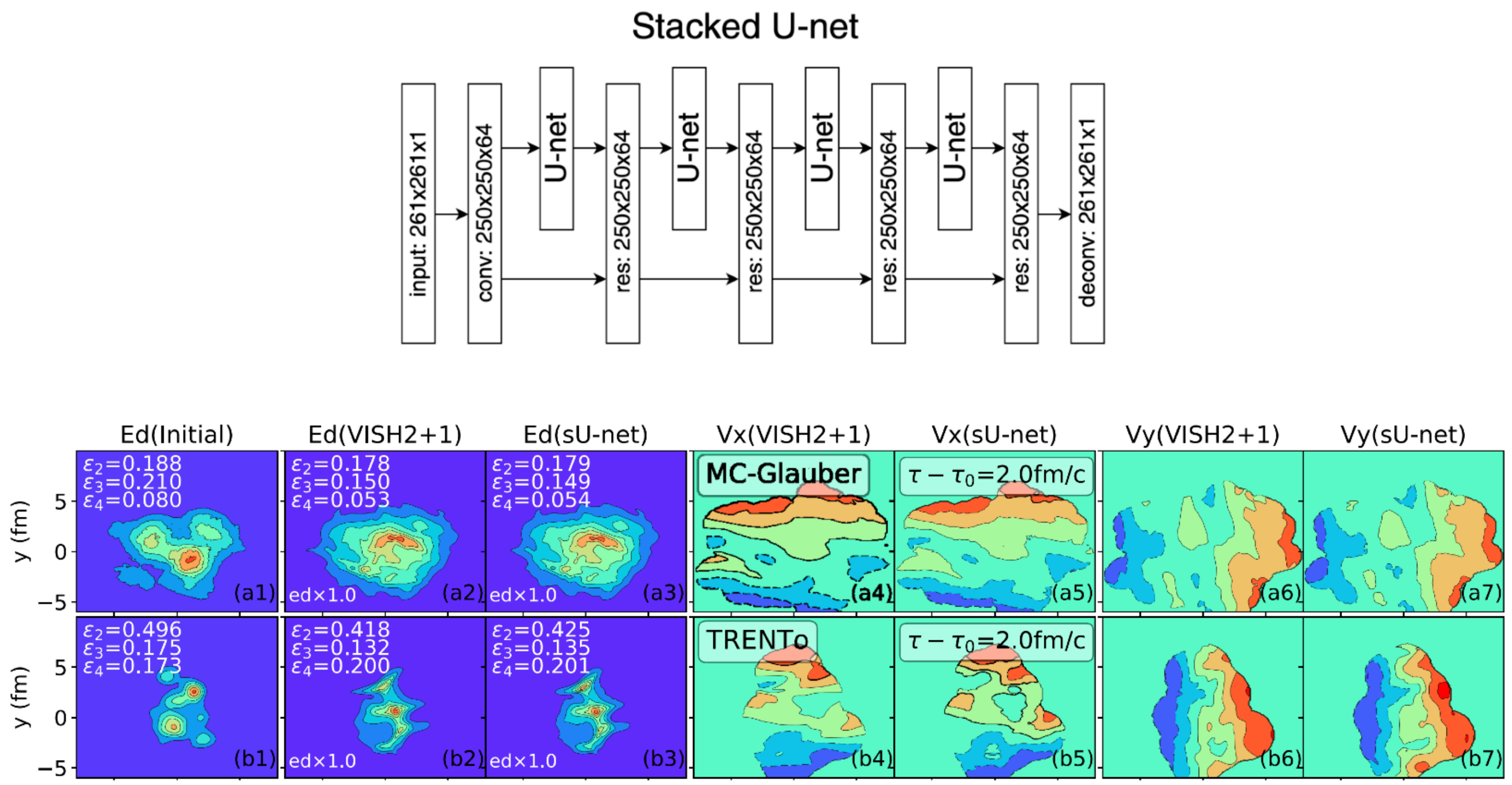
R. Wang, Y.-G. Ma, R. Wada, L.-W. Chen, W.-B. He, H.-L. Liu, and K.-J. Sun, Phys. Rev. Res. **2**, 043202 (2020)

Recognizing nuclear liquid-gas phase transition



R. Wang, Y.-G. Ma, R. Wada, L.-W. Chen, W.-B. He, H.-L. Liu, and K.-J. Sun,

Predict relativistic hydrodynamics



H. Huang, B. Xiao, Z. Liu, Z. Wu, Y. Mu, and H. Song, Phys. Rev. Res. **3**, 023256 (2021)

Backups I

Reconstructing Spectral Function

Comput. Phys. Commun. 282, 108547

- In practice, the Euclidean correlations have **finite number of points** and **with finite precision**;
- The ill-posedness of the spectral reconstruction **fundamentally exists even for** *continuous correlation functions(infinite observations)*;
- It's caused by the **numerical inaccuracy** of the correlation measurements (induced high degeneracy in solution space).

$$K_{ij}, i \in N_x, j \in N_\omega, N_x < N_\omega$$

$$\vec{D} \equiv K\vec{\rho}$$

highly rectangular

↑ vectorization

$$D(x) \equiv \int_0^\infty K(x,\omega)\rho(\omega)\mathrm{d}\omega$$

↓ eigenvalue problem

$$\int_0^\infty \psi_s(\omega)\, K(x,\omega)\, \mathrm{d}\omega = \lambda_s \psi_s(x)$$

J. Phys. A: Math. Gen., Vol. 11, No. 9, 1978. Printed in Great Britain.

Backups II: Autoregressive Networks

$$\max_{\theta} \prod_{i=1}^{N} p_\theta(\mathbf{s}_i)$$

arXiv:2007.01037

on specific degrees of freedom(*d.o.f.s*)

1. Prepare data-set from observations

$$\mathbf{s} \sim q_{\text{data}}$$

2. Put them into the deep autoregressive network(DAN)

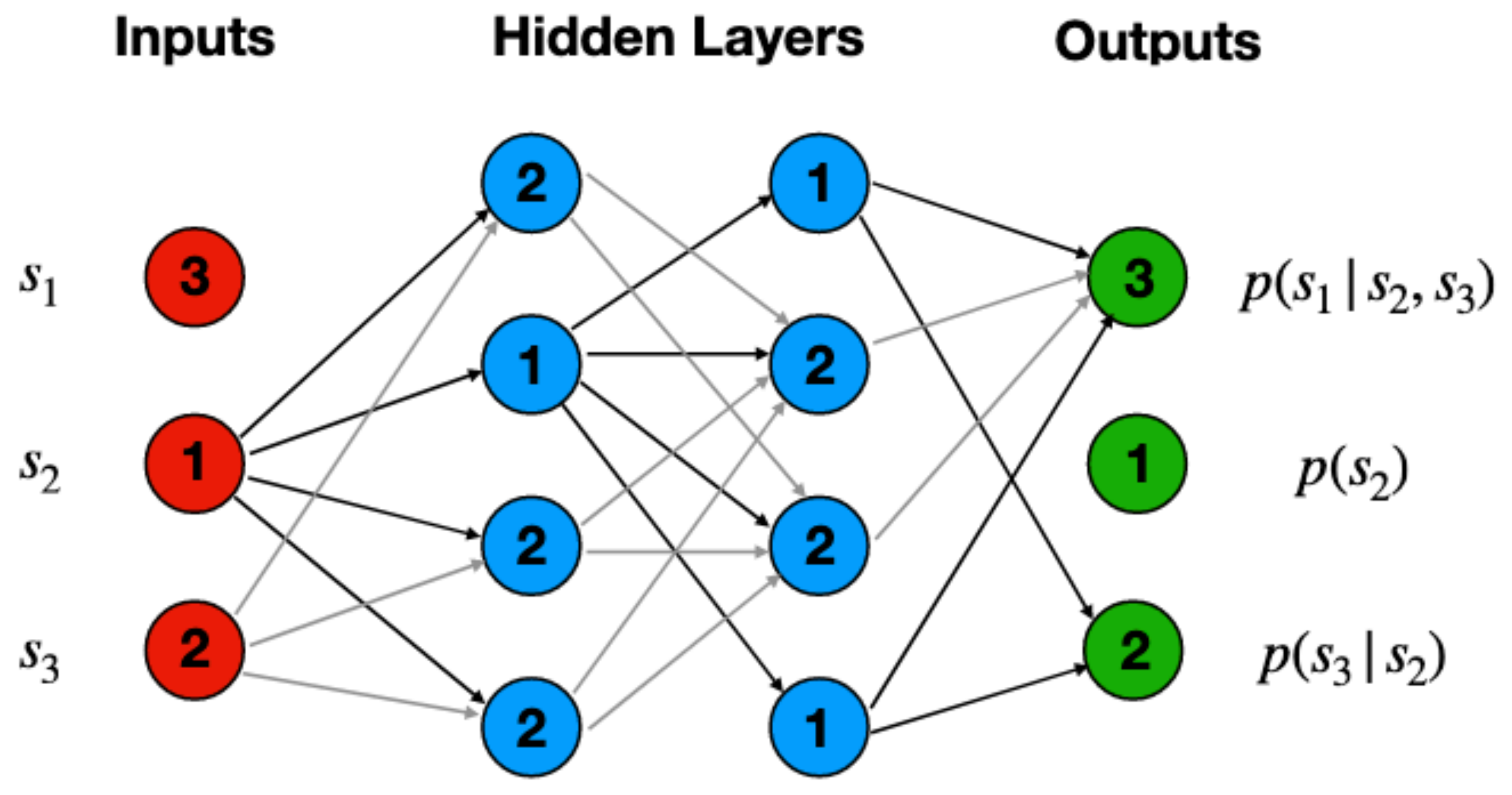
$$p_\theta(\mathbf{s}) = \prod_{i=1}^{N} p(s_i \,|\, s_1, \ldots, s_{i-1}, \theta)$$

3. Minimize the negative log-liklihood(NLL)

$$\mathscr{L} = -\sum_{\mathbf{s}\sim q_{data}} \sum_{d=1}^{N} \log(p(s_d \,|\, \mathbf{s}_{<d}, \theta))$$

4. Get your DAN represented Hamiltonian

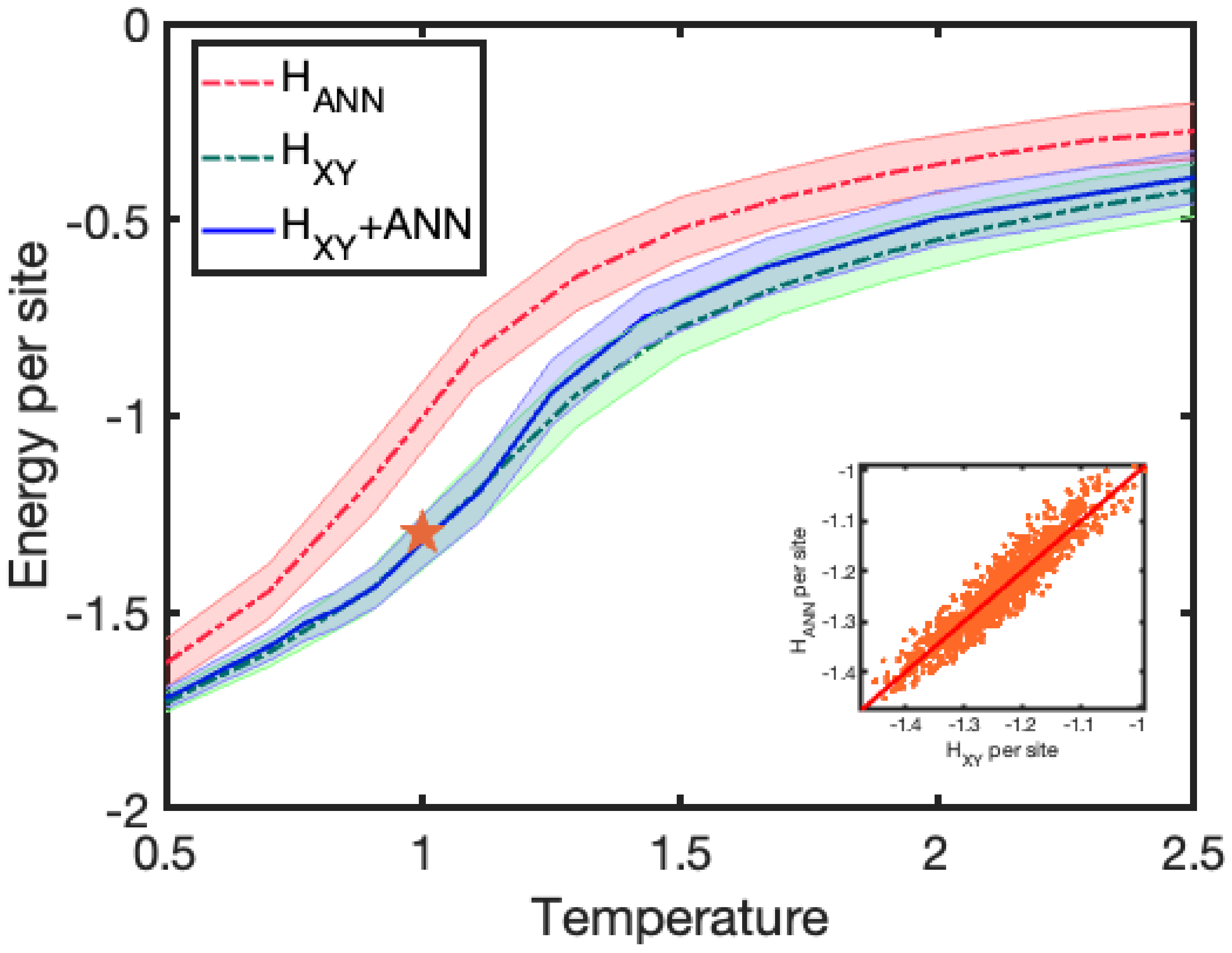
$$H_\theta(\mathbf{s}, T) = -T \ln p_\theta(\mathbf{s})$$



Backups II: Topological Phase Transition

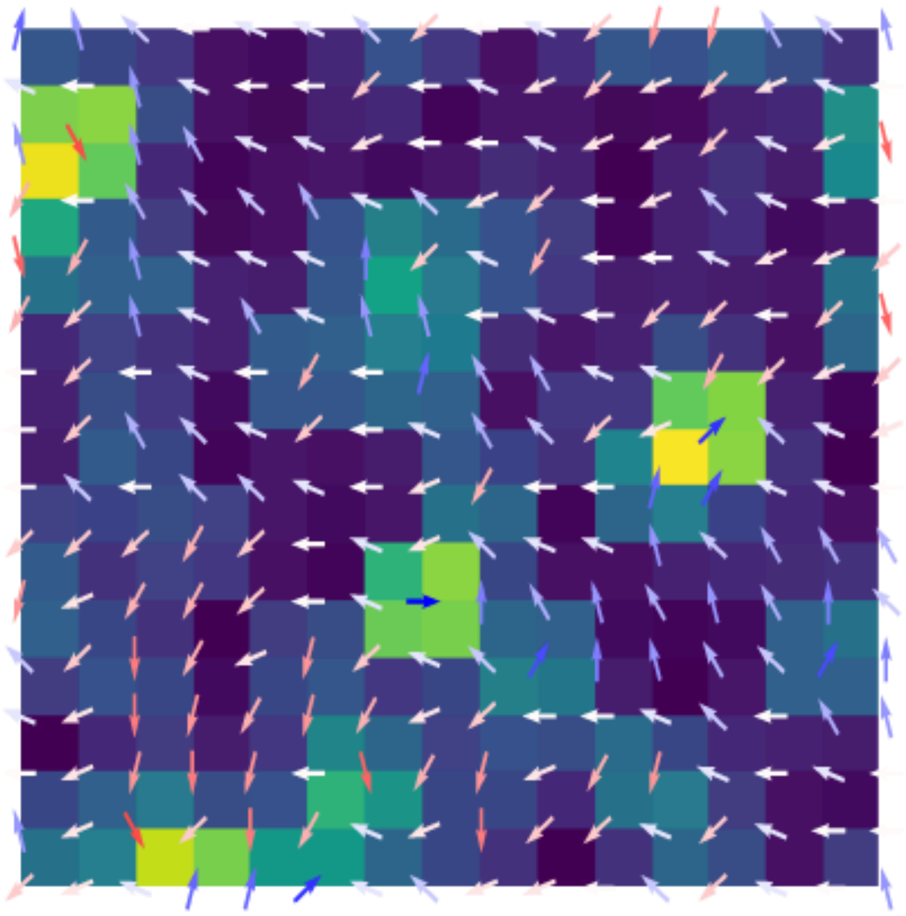
2D XY Model

$$H_{\rm XY} = -\sum_{<i,j>} s_i s_j = -\sum_{<i,j>} \cos(\phi_i - \phi_j)$$

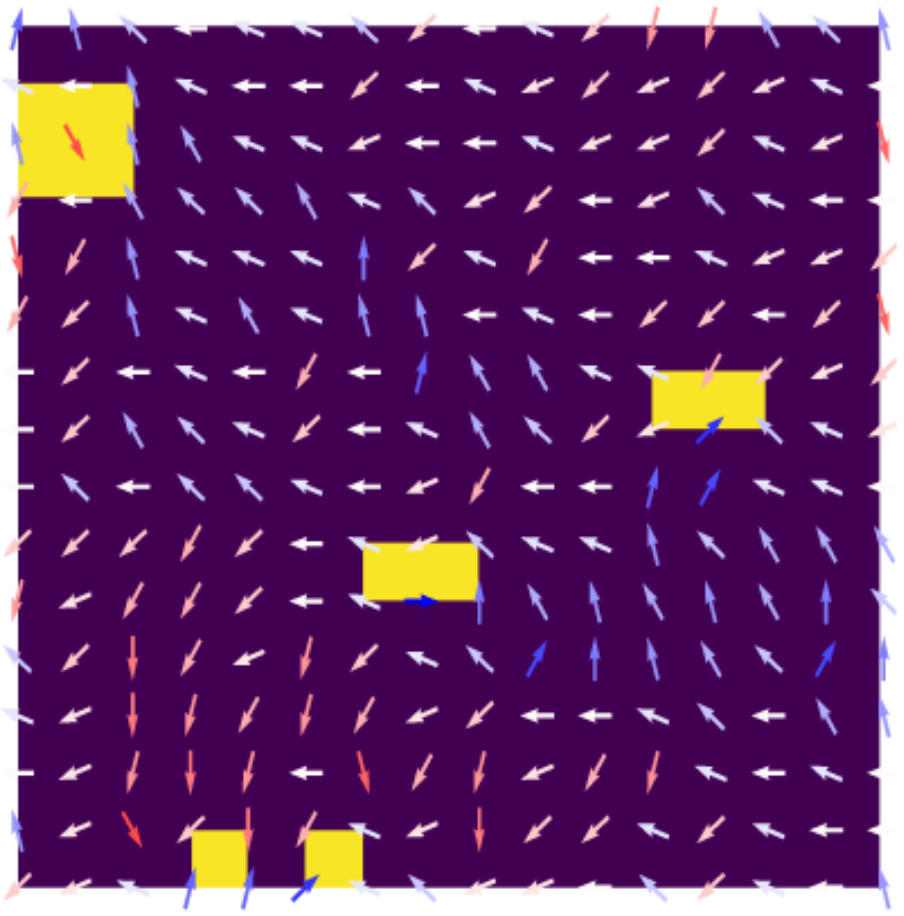


9000 training configurations with L = 16

Chinese Phys. Lett. 39, 120502 (2022)



Probability distributions from CANs



Vortices

$$\delta q_{[i,j]} \equiv \sum_{[i,j]} q_\theta(s_{ij})$$

$$v = (1/2\pi) \oint_C \nabla\phi(\boldsymbol{r}) \cdot d\boldsymbol{r}$$

$q_\theta(s_{ij})$ **conditional probability differences in the same given direction**

Adding noise

$$\mathbf{x} \sim p_{data}(\mathbf{x}) \equiv p(\mathbf{x})$$

$$\mathbb{E}_{p(\mathbf{x})}[||\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})||_2^2]$$

Minimizing the Fisher divergence

1-D

$$\tilde{x} = x + \sigma\epsilon \qquad p_\sigma(\tilde{x}) \equiv \int p_\sigma(\tilde{x}|x) p_{data}(x) dx$$

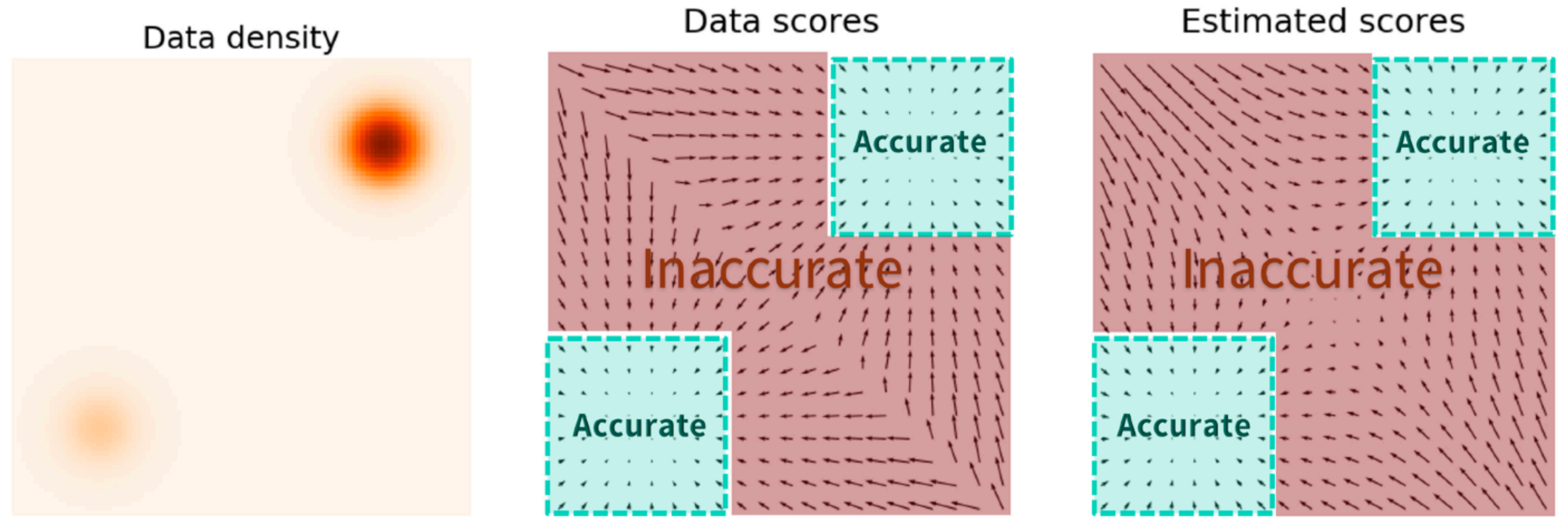
$$p_\sigma(\tilde{x}|x) \sim N(\tilde{x}; x, \sigma^2 I), \epsilon \sim N(0, I)$$

$$\frac{\partial \log(p_\sigma(\tilde{x}))}{\partial \tilde{x}} \equiv \frac{\partial \log(p_\sigma(\tilde{x}|x) p_{data}(x))}{\partial \tilde{x}} = \frac{\partial \log(p_\sigma(\tilde{x}|x))}{\partial \tilde{x}}$$

Score-matching a noise-perturbed distribution

$$\mathbb{E}_{p_\sigma(\mathbf{x})}[||\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})||_2^2]$$

Matching a noise-perturbed distribution



Estimated scores are only accurate in high density regions

No enough data in low-density region!
But we sample from the low-density region using Langevin dynamics...

Diffusion Models

$$\mathbb{E}_{p_\sigma(\mathbf{x})}[\|\nabla_\mathbf{x} \log p_\sigma(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$$

Matching a noise-perturbed distribution

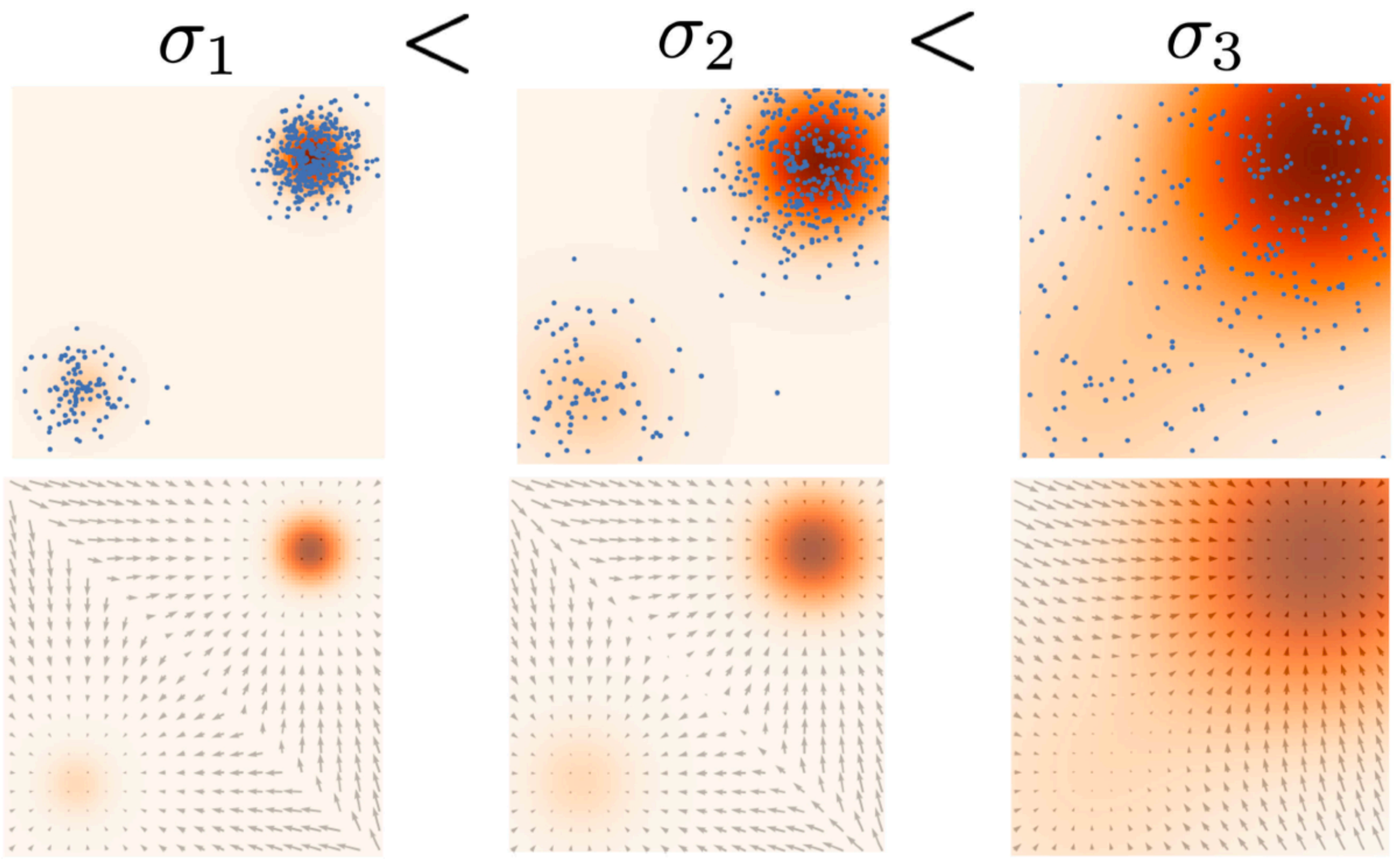
Small noise → Good approximation to data, poor in low-density region!

Large noise → Poor approximation to data, good in low-density region!

$$\sum_{i=1}^{L} \lambda_i \mathbb{E}_{p_{\sigma_i}(\mathbf{x})}[\|\nabla_\mathbf{x} \log p_{\sigma_i}(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, i)\|_2^2]$$

Matching multi-noise perturbed distributions

Multiple noise perturbations!



- Choose the noise scheme as a geometric progression, $\sigma_1/\sigma_2 = \sigma_{i-1}/\sigma_i = \ldots = \sigma_{L-1}/\sigma_L > 1$, with σ_1 **being sufficiently small** and σ_L **comparable to the maximum pairwise distance between all training data points**. L is typically on the order of hundreds or thousands.
- Parameterize the score-based model, $\mathbf{s}_\theta(\mathbf{x}, i)$, with U-Net skip connections.

Multiple scales of Gaussian noise to perturb the data distribution (first row), and jointly estimate the score functions for all of them (second row).

Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020

Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Diffusion Models

$$\sum_{i=1}^{L} \lambda_i \mathbb{E}_{p_{\sigma_i}(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}, i)\|_2^2]$$

Matching multi-noise perturbed distributions

$$x_{i+1} = x_i + \sigma_i \epsilon$$

$$\frac{d\mu}{dt} = 0$$

$$\frac{d\sigma^2}{dt} = \sigma^{2t}$$

$t \in [0,\tau]$

$$\nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}) = \nabla_{\mathbf{x}} \log(p_{\sigma_i}(\mathbf{x}_\tau | \mathbf{x}) p(\mathbf{x})) = \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}_\tau | \mathbf{x})$$

$$p_{\sigma_i}(\mathbf{x}_\tau \mid \mathbf{x}) = \mathcal{N}\left(\mathbf{x}_\tau; \mathbf{x}, \frac{1}{2\log\sigma}(\sigma^{2\tau} - 1)\mathbf{I}\right)$$

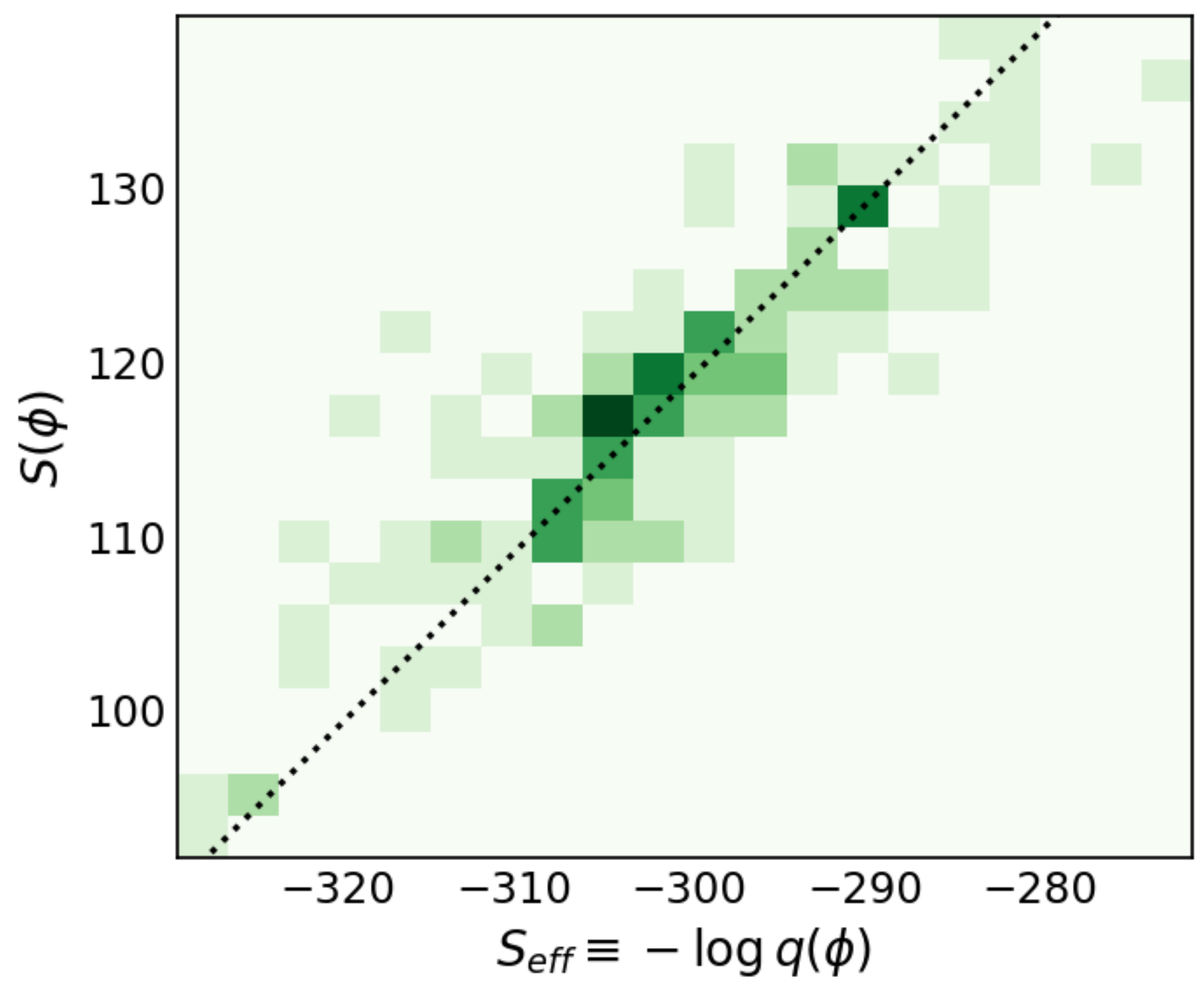
- **Choose the noise scheme as, $\sigma_i \equiv \sigma^\tau$, where τ indicates "time-step" for adding noise.**

H. Risken, The Fokker-Planck Equation: Methods of Solution and Applications

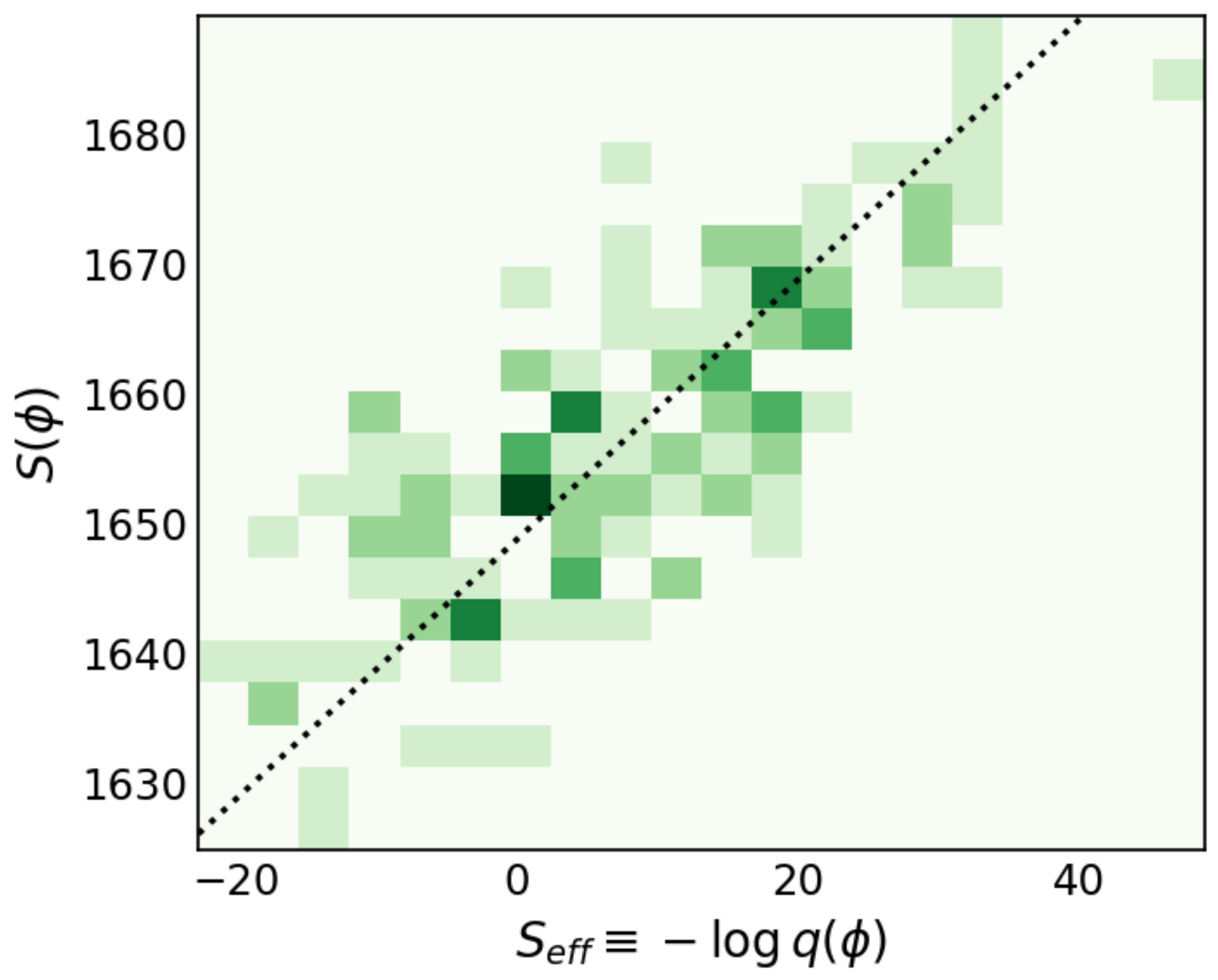
Training is Matching

Backups IIII: Likelihood in DM

$\beta = 1$



$\beta = 7$



Learned at $\beta = 1$ with **10,240** cofigurations, L = 16
Generated 128 configs for estimating Likelihood