Machine Learning for QCD Matter: from Inverse Problems to Generative Models

Lingxiao Wang(王凌霄) (RIKEN)

May 18, 2024
The 1st edition of Spicy Gluons Workshop for Young Scientists

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.

Advertisement

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 -)

DEEP-IN Working Group Website

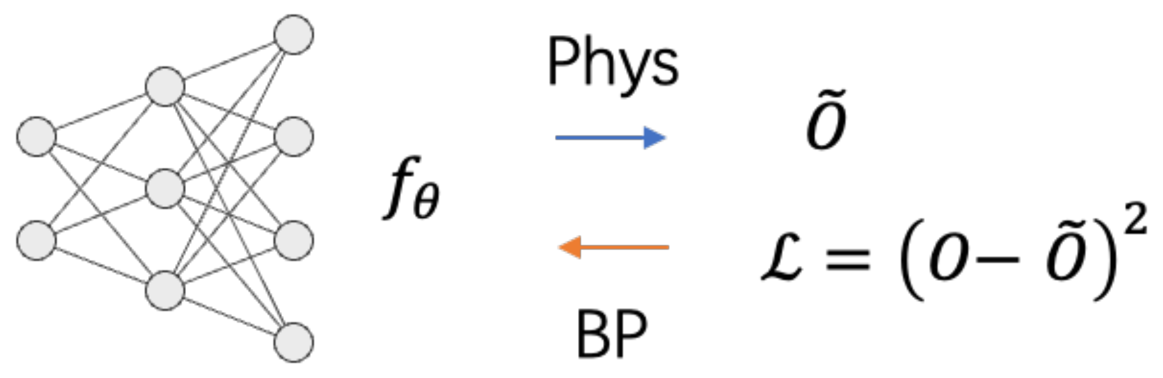
Objectives

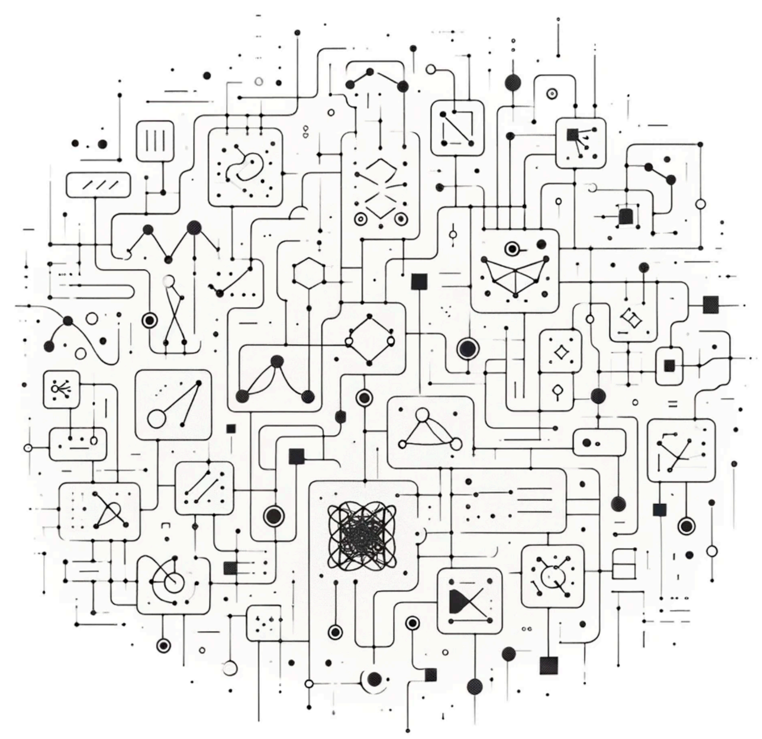
The essence of discovery in sciences has always been rooted in the reverse engineering of natural phenomena and observational data. This paradigm of deducing the underlying laws of nature from observable outcomes forms the cornerstone of our scientific inquiry. The DEEP-IN working group is established with the recognition that the elucidation of such complex phenomena demands a fusion of physics insights and advanced deep learning methodologies. Historically, the exploration of sceinces has relied heavily on intuition and empirical exploration, with methods like inference playing a significant role in our understanding.

Facilitators:
Lingxiao Wang (RIKEN iTHEMS) *Contact at lingxiao.wang@riken.jp
Catherine Beauchemin (RIKEN iTHEMS)
Akinori Tanaka (RIKEN AIP/RIKEN iTHEMS)
Enrico Rinaldi (Quantinuum K.K./RIKEN iTHEMS)
Akira Harada (NITIC/RIKEN iTHEMS)

Outline

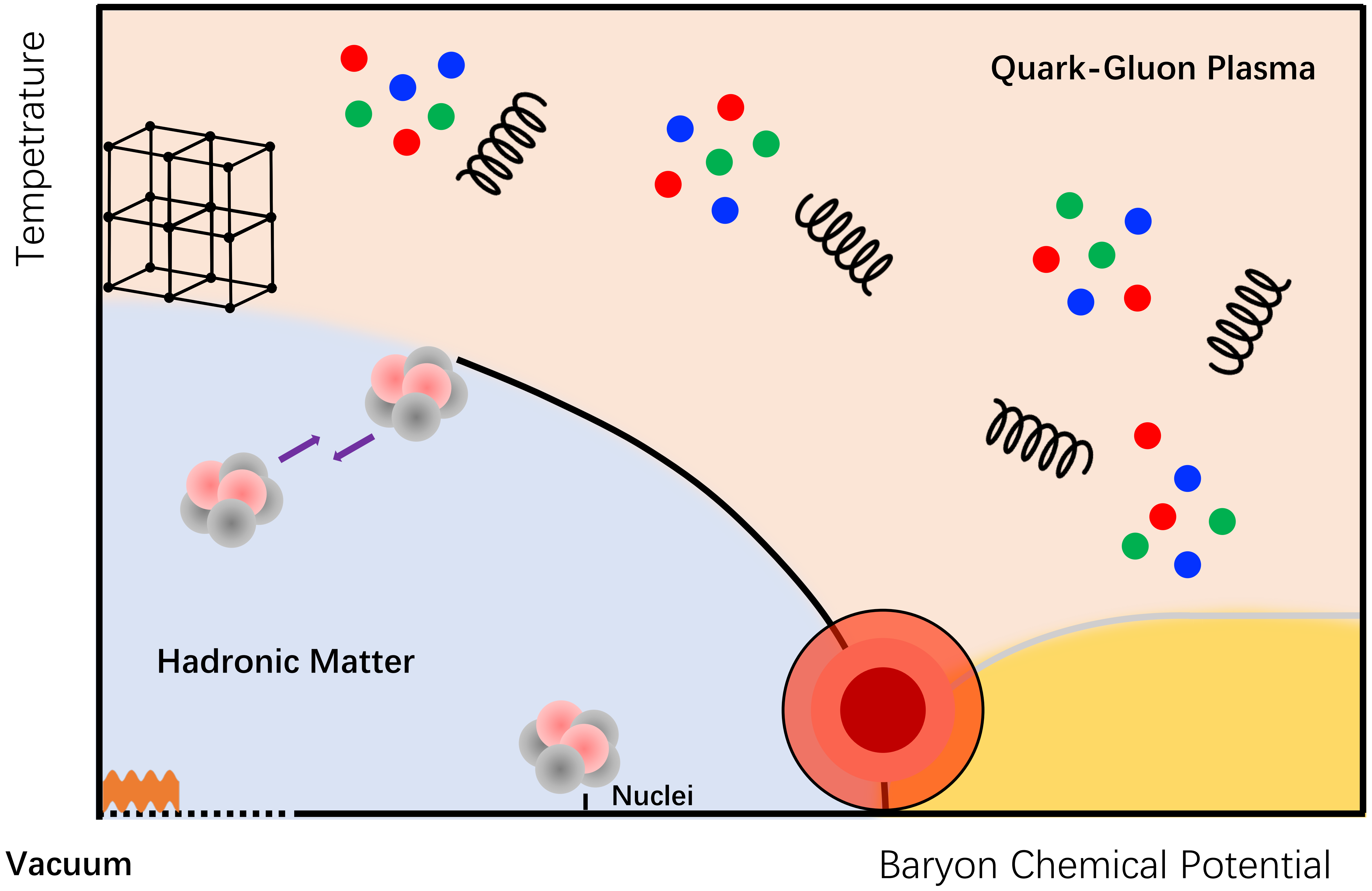
- **Machie Learning for QCD Matter**
- **Inverse Problems**
  - Data-Driven Learning
  - Physics-Driven Learning
- **Generative Models**
  - Probabilistic Models
  - Diffusion Models
- **Outlooks**

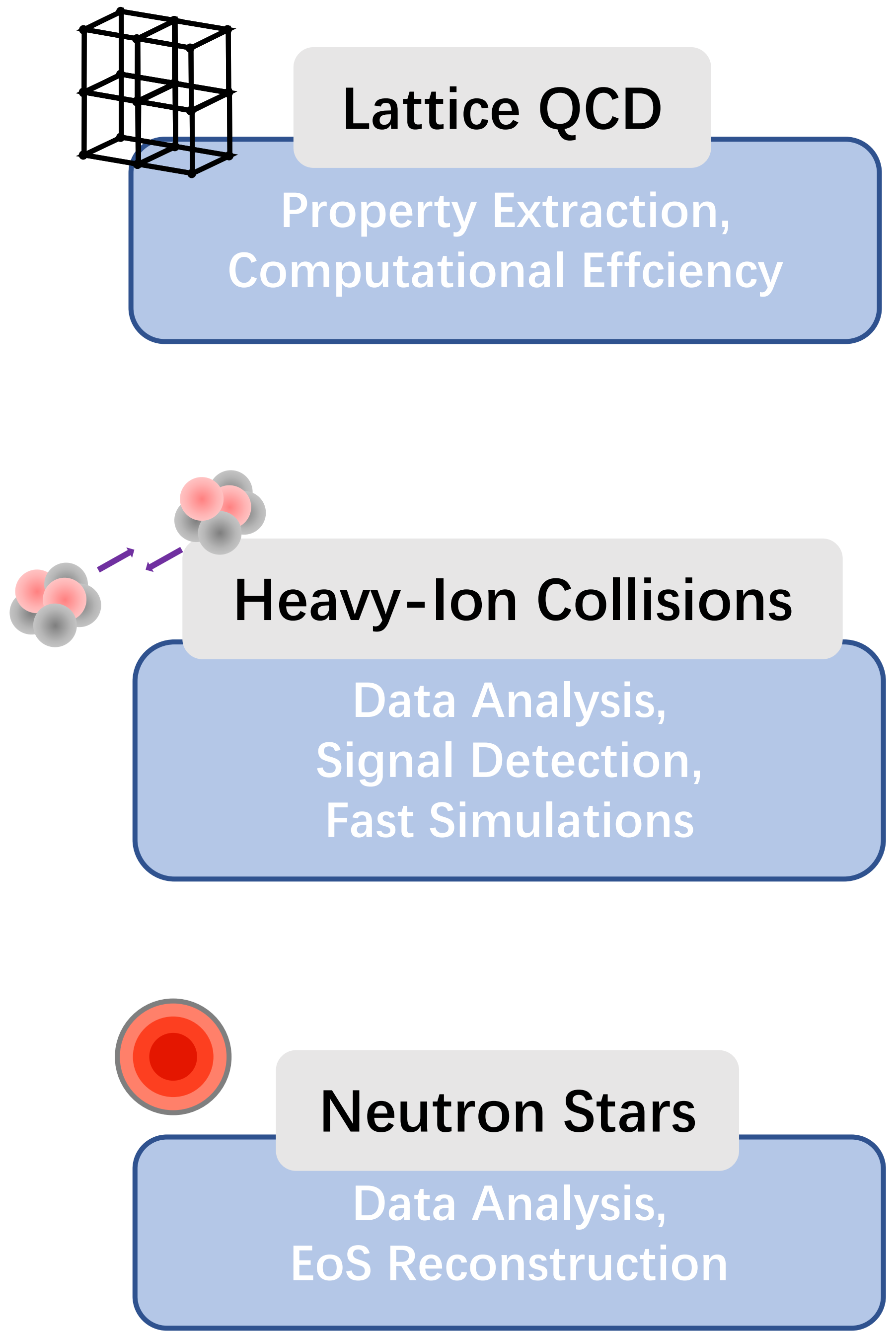




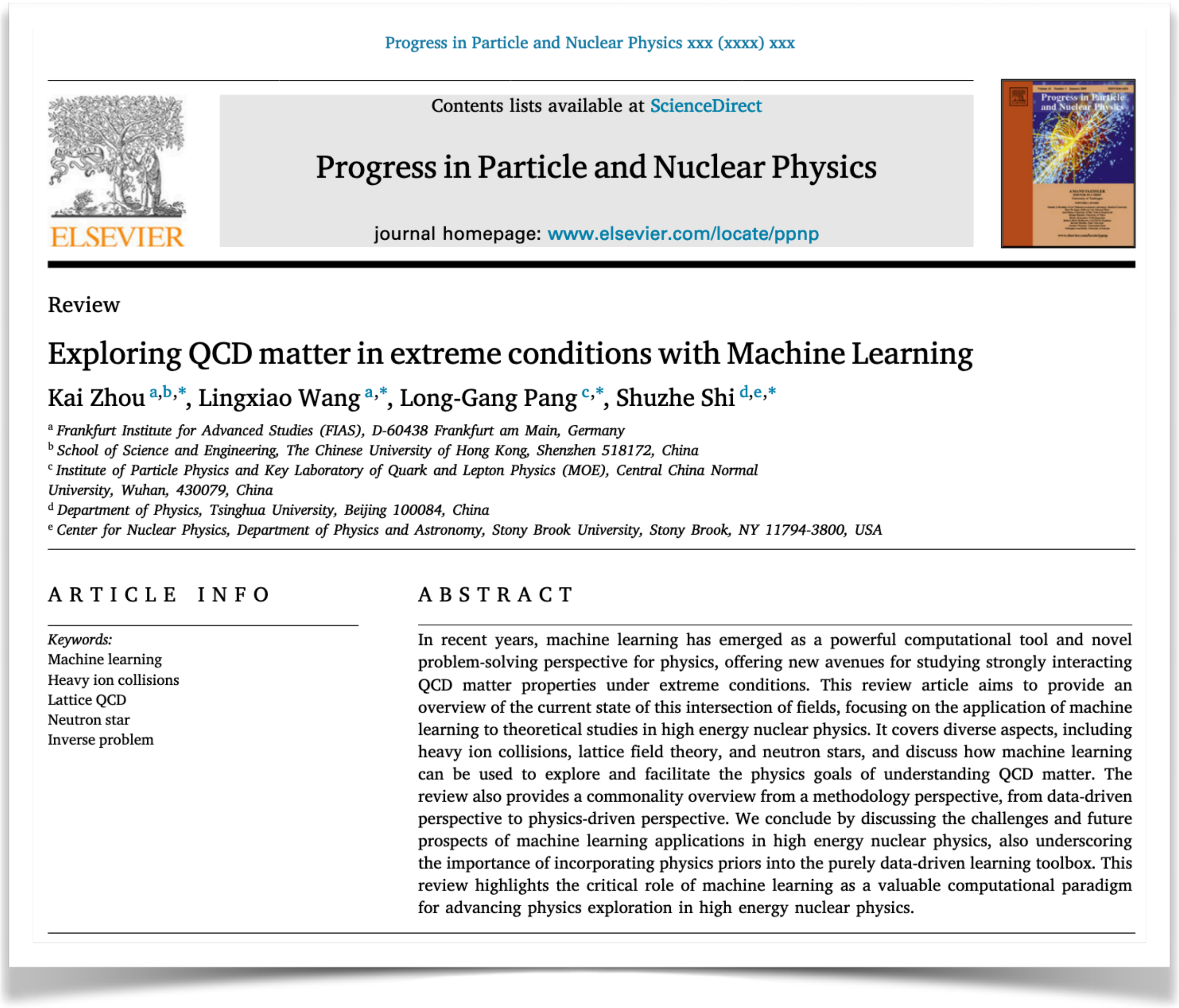
Generated by ChatGPT-4 + DALL·E

Why Machine Learning?





Prog.Part.Nucl.Phys. 104084(2023) **(Invited Review)**



Progress in Particle and Nuclear Physics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Progress in Particle and Nuclear Physics

journal homepage: www.elsevier.com/locate/ppnp

Review

Exploring QCD matter in extreme conditions with Machine Learning

Kai Zhou [a,b,*], Lingxiao Wang [a,*], Long-Gang Pang [c,*], Shuzhe Shi [d,e,*]

[a] Frankfurt Institute for Advanced Studies (FIAS), D-60438 Frankfurt am Main, Germany
[b] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China
[c] Institute of Particle Physics and Key Laboratory of Quark and Lepton Physics (MOE), Central China Normal University, Wuhan, 430079, China
[d] Department of Physics, Tsinghua University, Beijing 100084, China
[e] Center for Nuclear Physics, Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794-3800, USA

ARTICLE INFO

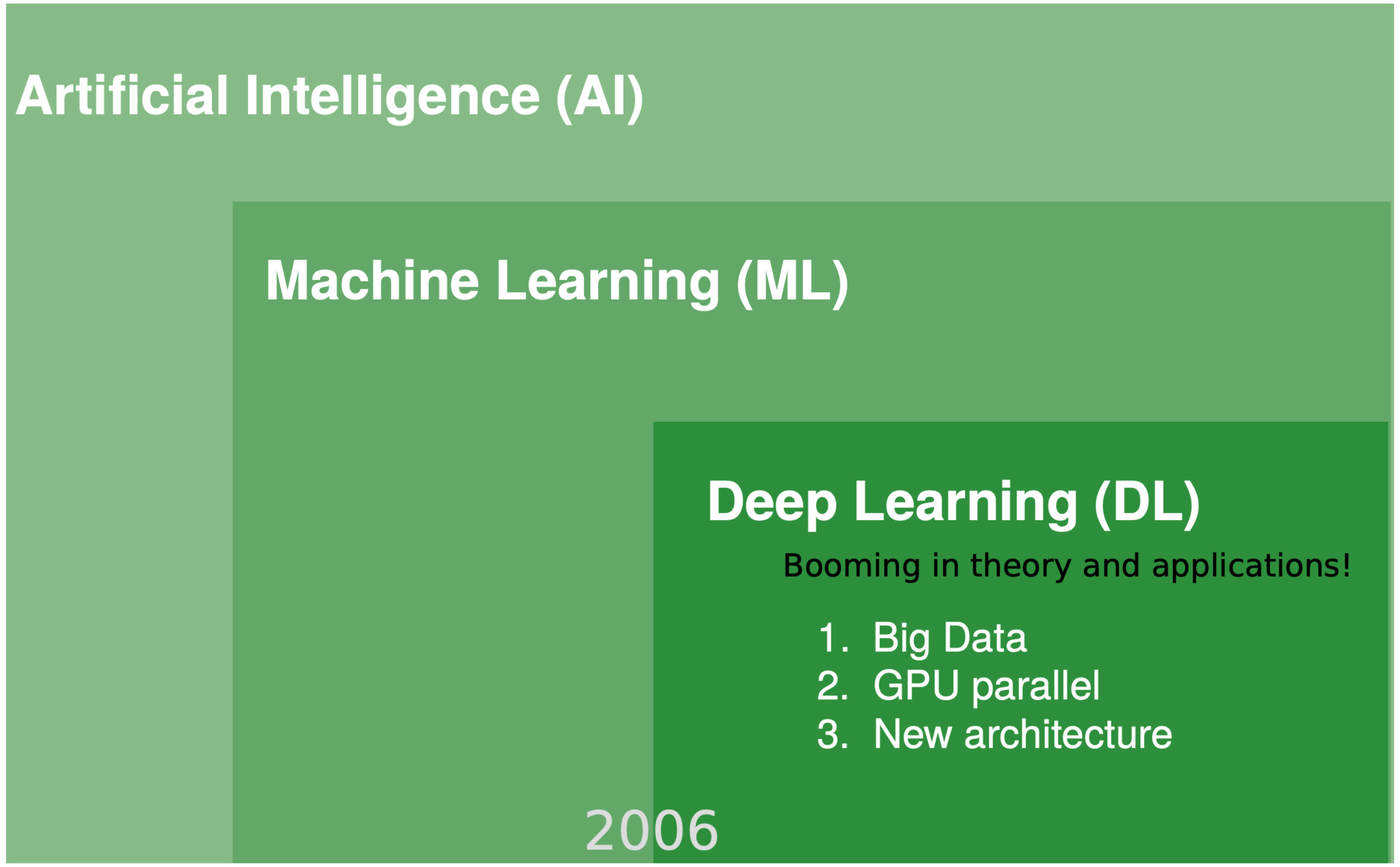
Keywords:
Machine learning
Heavy ion collisions
Lattice QCD
Neutron star
Inverse problem

ABSTRACT

In recent years, machine learning has emerged as a powerful computational tool and novel problem-solving perspective for physics, offering new avenues for studying strongly interacting QCD matter properties under extreme conditions. This review article aims to provide an overview of the current state of this intersection of fields, focusing on the application of machine learning to theoretical studies in high energy nuclear physics. It covers diverse aspects, including heavy ion collisions, lattice field theory, and neutron stars, and discuss how machine learning can be used to explore and facilitate the physics goals of understanding QCD matter. The review also provides a commonality overview from a methodology perspective, from data-driven perspective to physics-driven perspective. We conclude by discussing the challenges and future prospects of machine learning applications in high energy nuclear physics, also underscoring the importance of incorporating physics priors into the purely data-driven learning toolbox. This review highlights the critical role of machine learning as a valuable computational paradigm for advancing physics exploration in high energy nuclear physics.

- **Heavy-Ion Collisions** : **Large number of data!** Complicated simulations!
- **Neutron Star** : **Accumulating observations!** Poor signal-noise ratio!
- **Lattice QCD** : **Computationally consuming!** Physics extraction!

What is ML?



Geoffrey Hinton

Machine Learning (ML) is a **subset of artificial intelligence** that **involves the creation of** algorithms that **allow computers to** learn **from and make decisions or predictions based on** data. It's essentially a way for computers to **"learn" from data** without being explicitly programmed to do so.

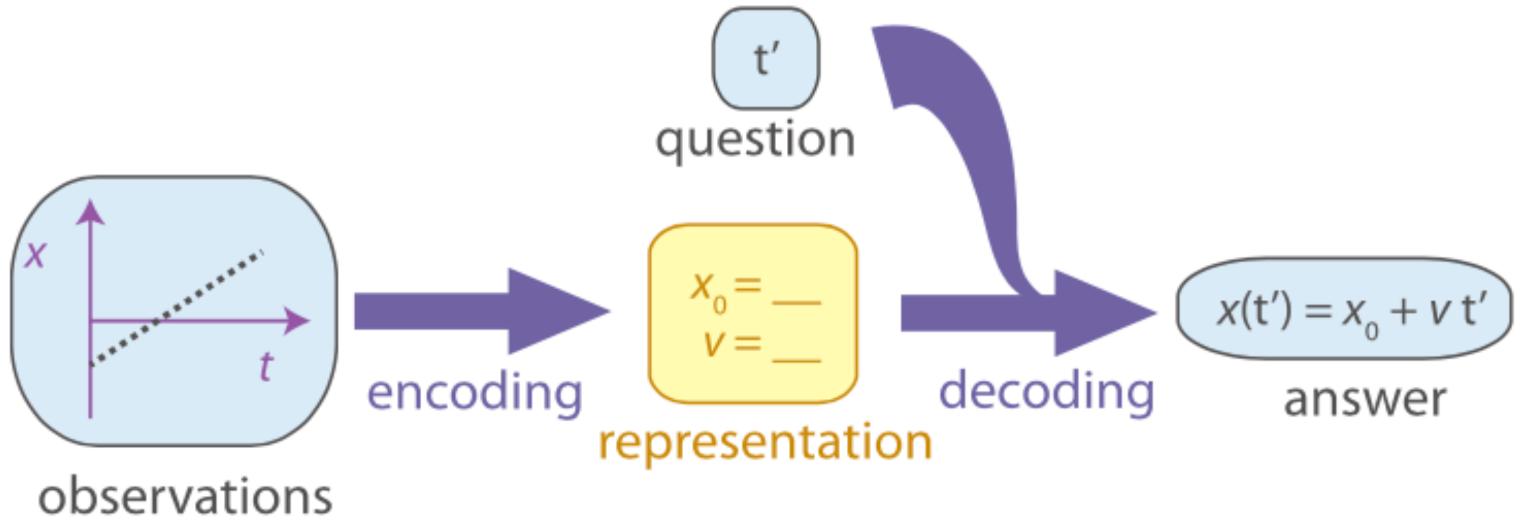
— **ChatGPT-4**

Big Data + Deep Models

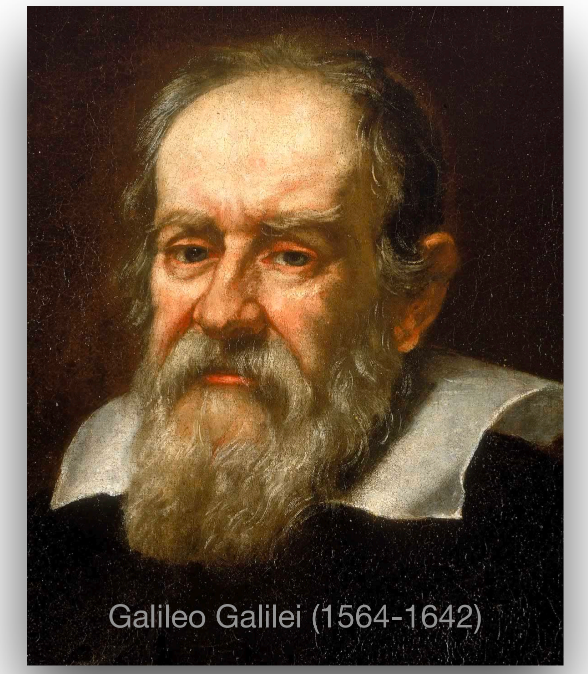
↓ **GPU** 

Successful Deep Learning!

Machine Learning and Physics

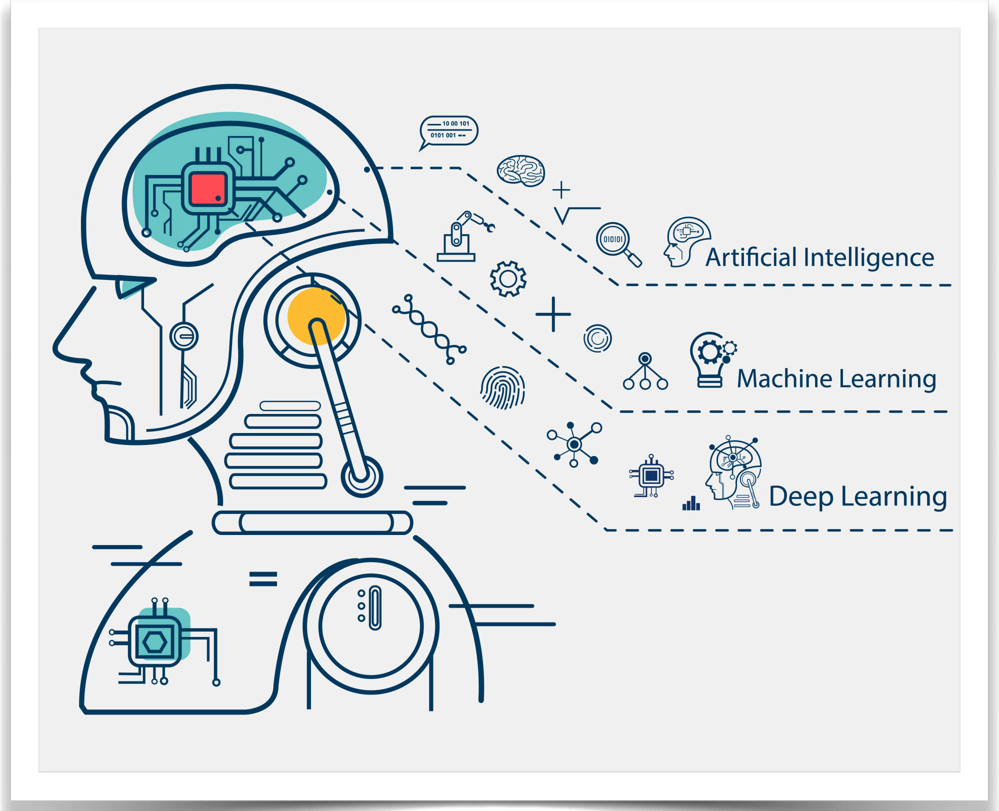


Phys.Rev. Lett. **124**, 010508 (2020)



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

Data, X



Machine, $\{\theta\}$

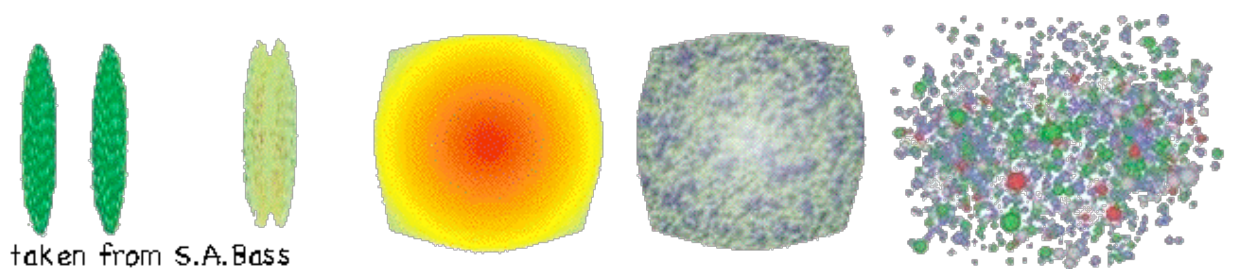
Prediction

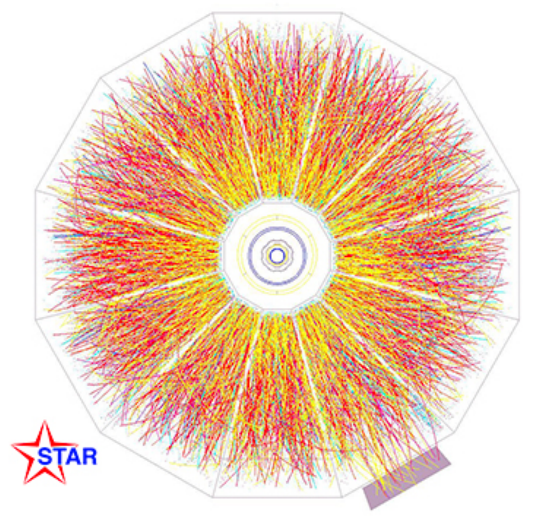
Estimation

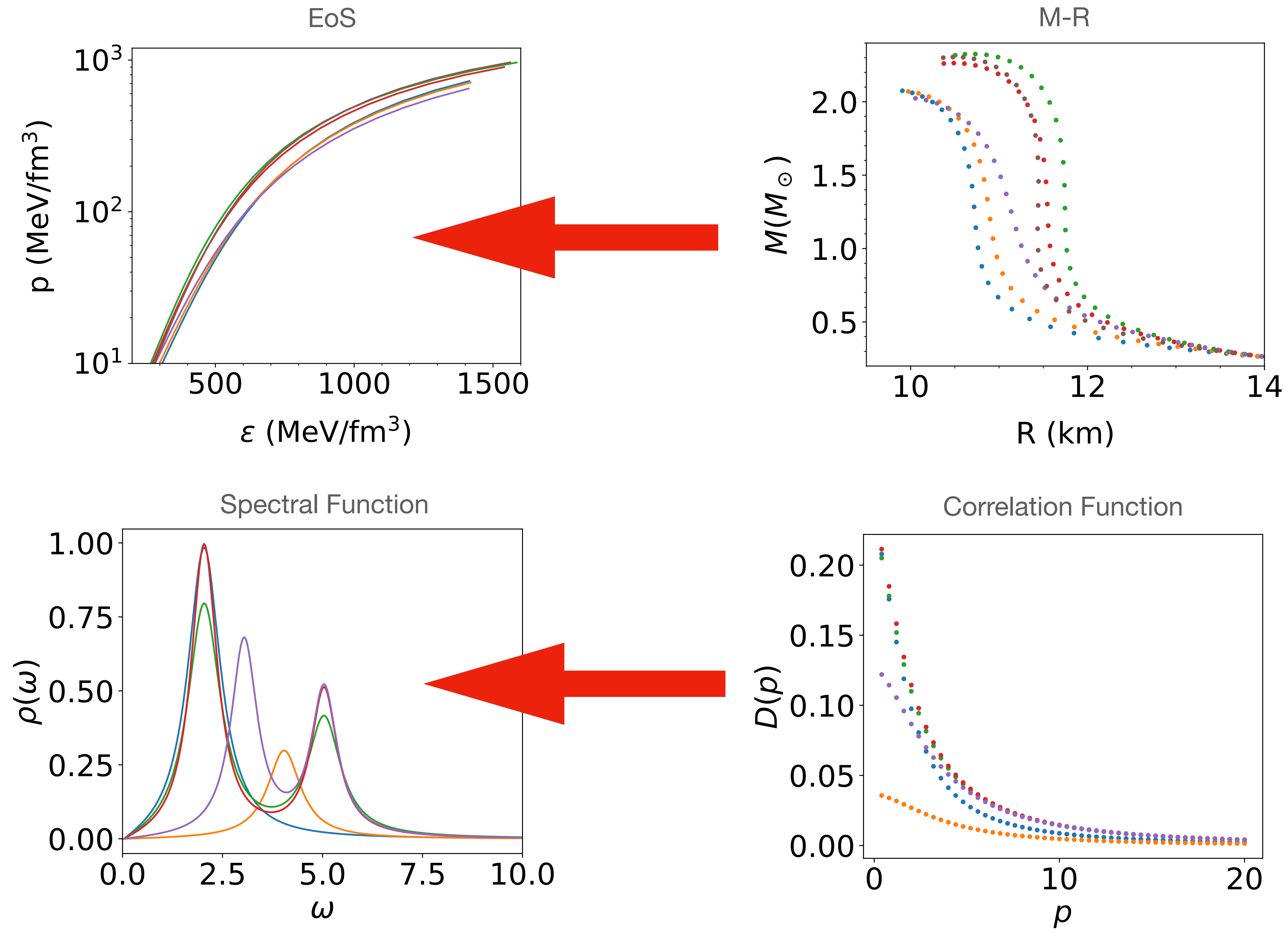
Inverse Problems

Physics

Data

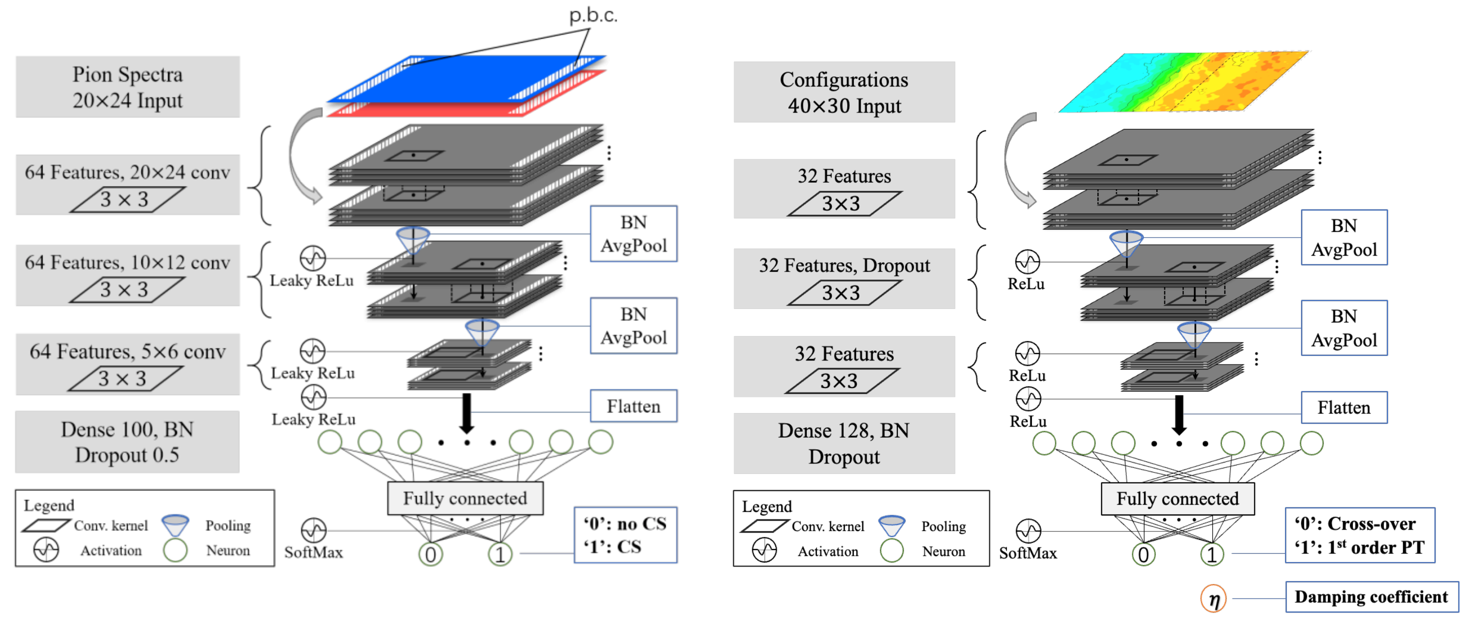






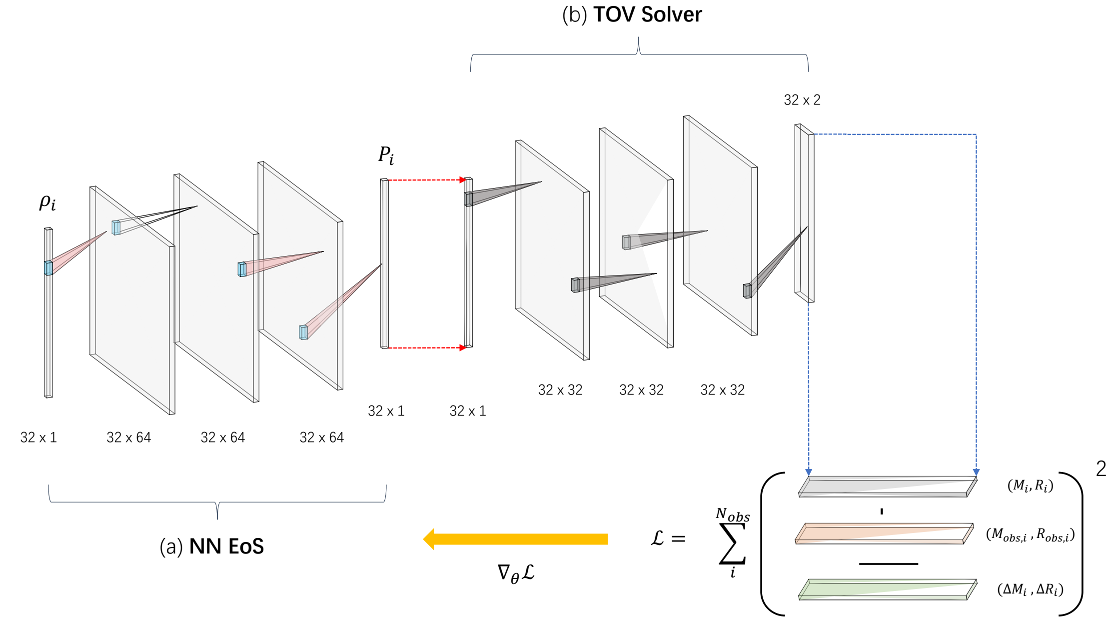
Heavy-Ion Collisions

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



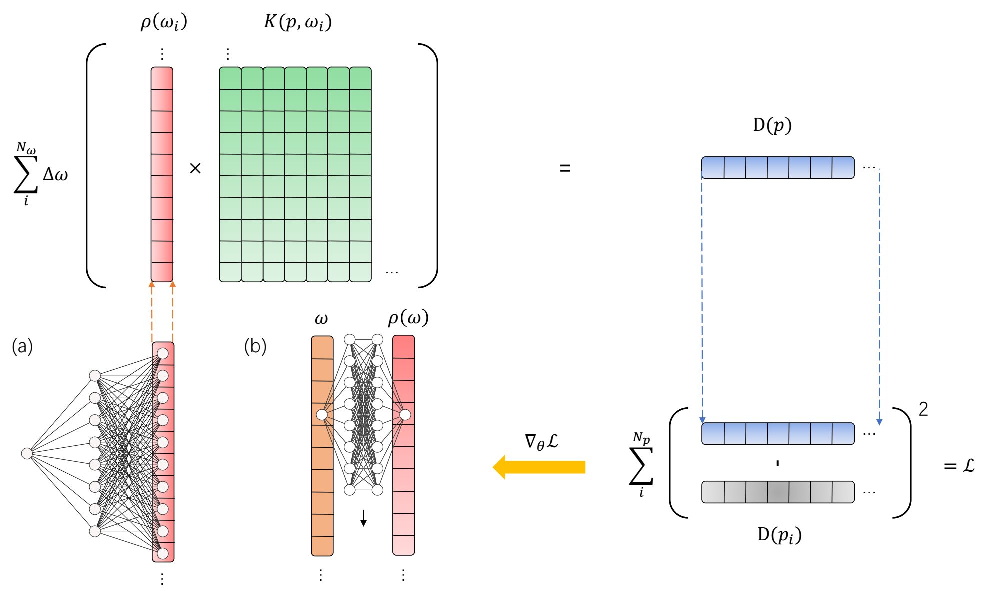
Neutron Star

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



Lattice QCD

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



Machine Learning and Inference

Maximum Likelihood Estimation(MLE)

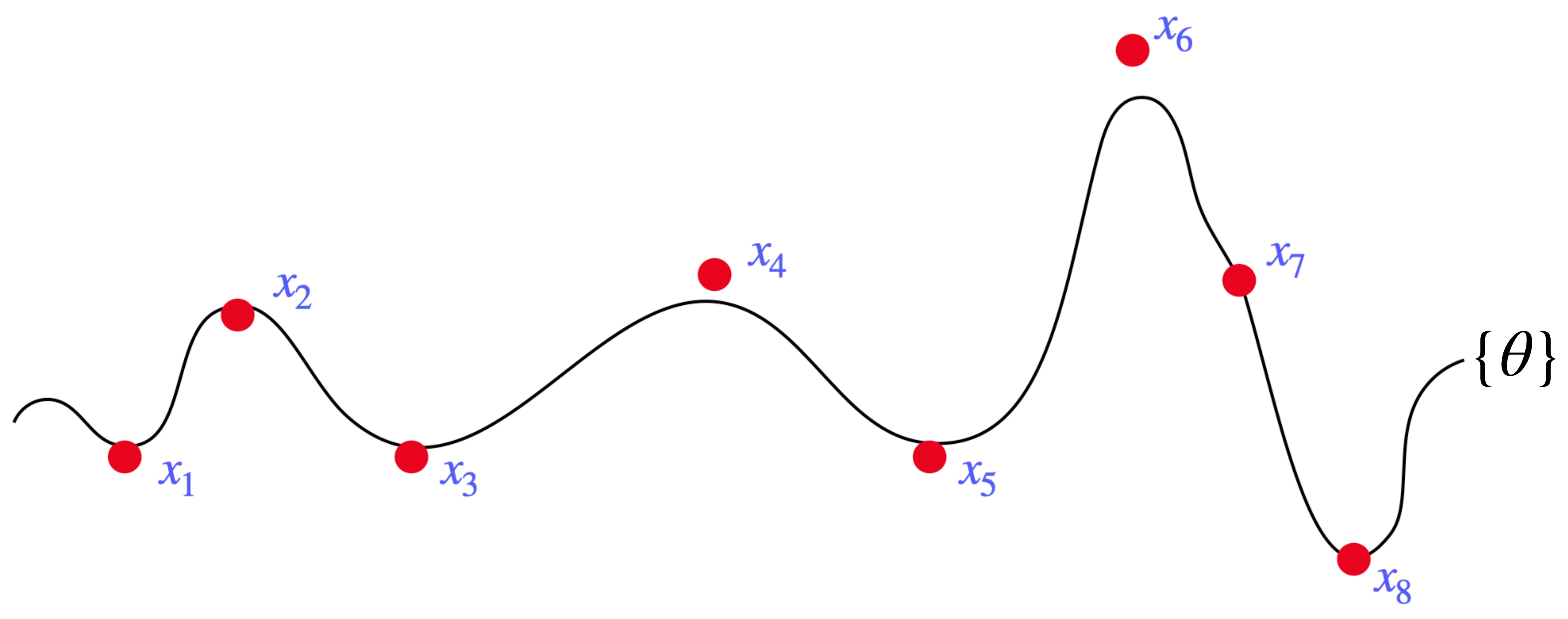
$$\max_{\theta} \prod_{i=1}^{N} p(\mathbf{x}_i \mid \theta)$$

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** $p(X)$



Data-Driven Learning

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

Forward process

Physics

Model Parameters/
Properties/States

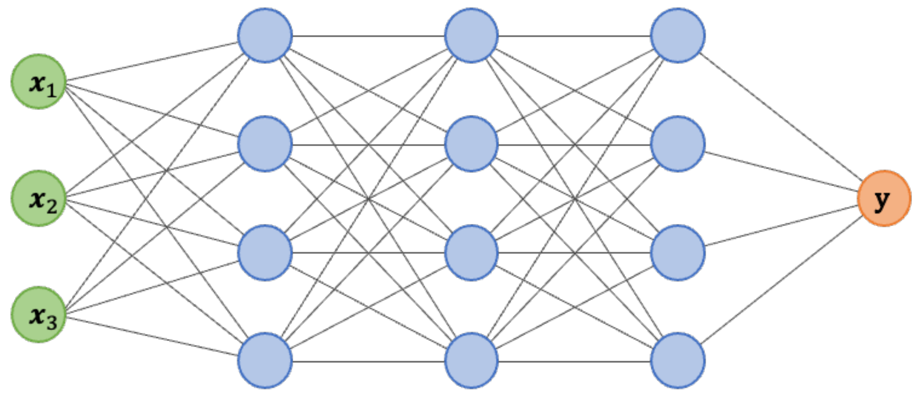
Data

Observations

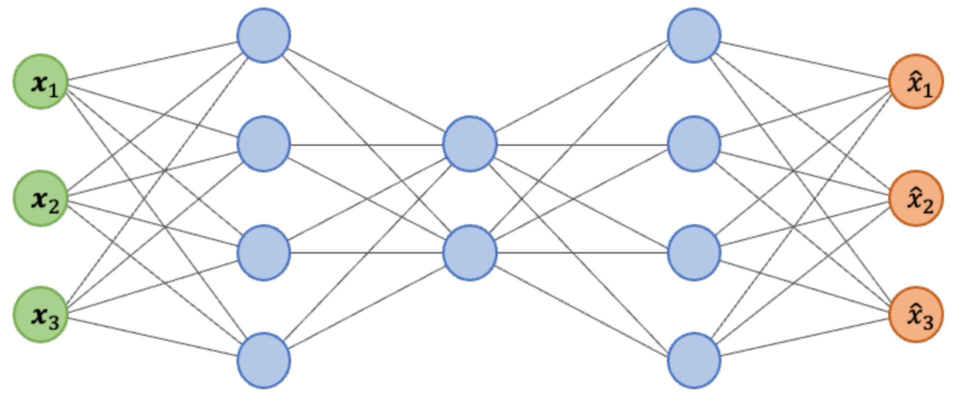
Inverse Mapping, f_θ

Data-Driven Learning

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



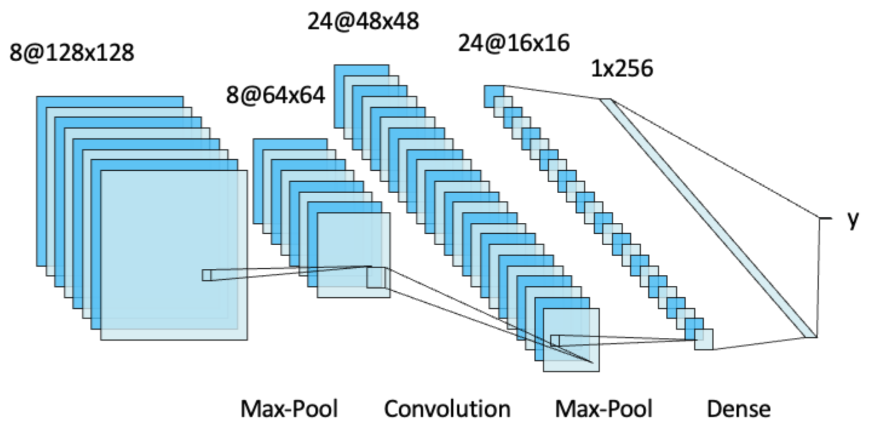
Deep Neural Network



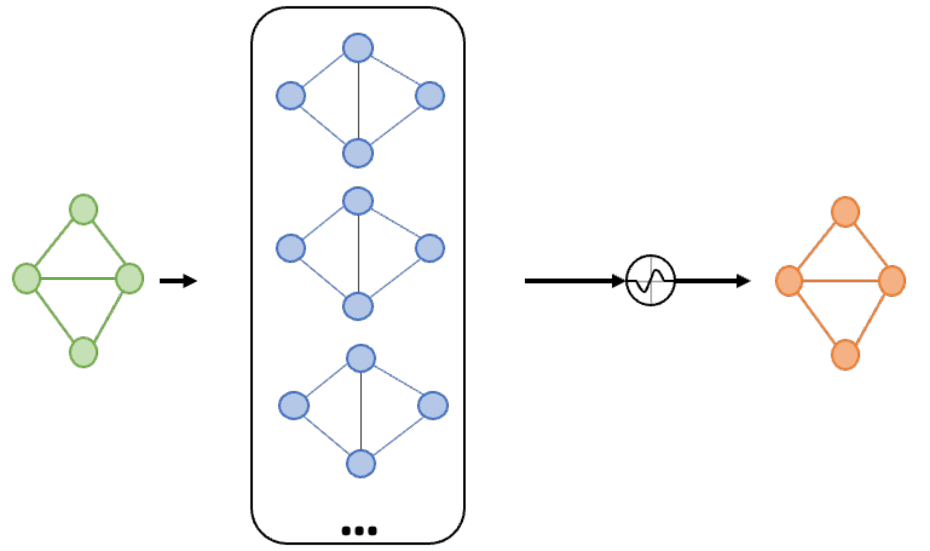
AutoEncoder

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.



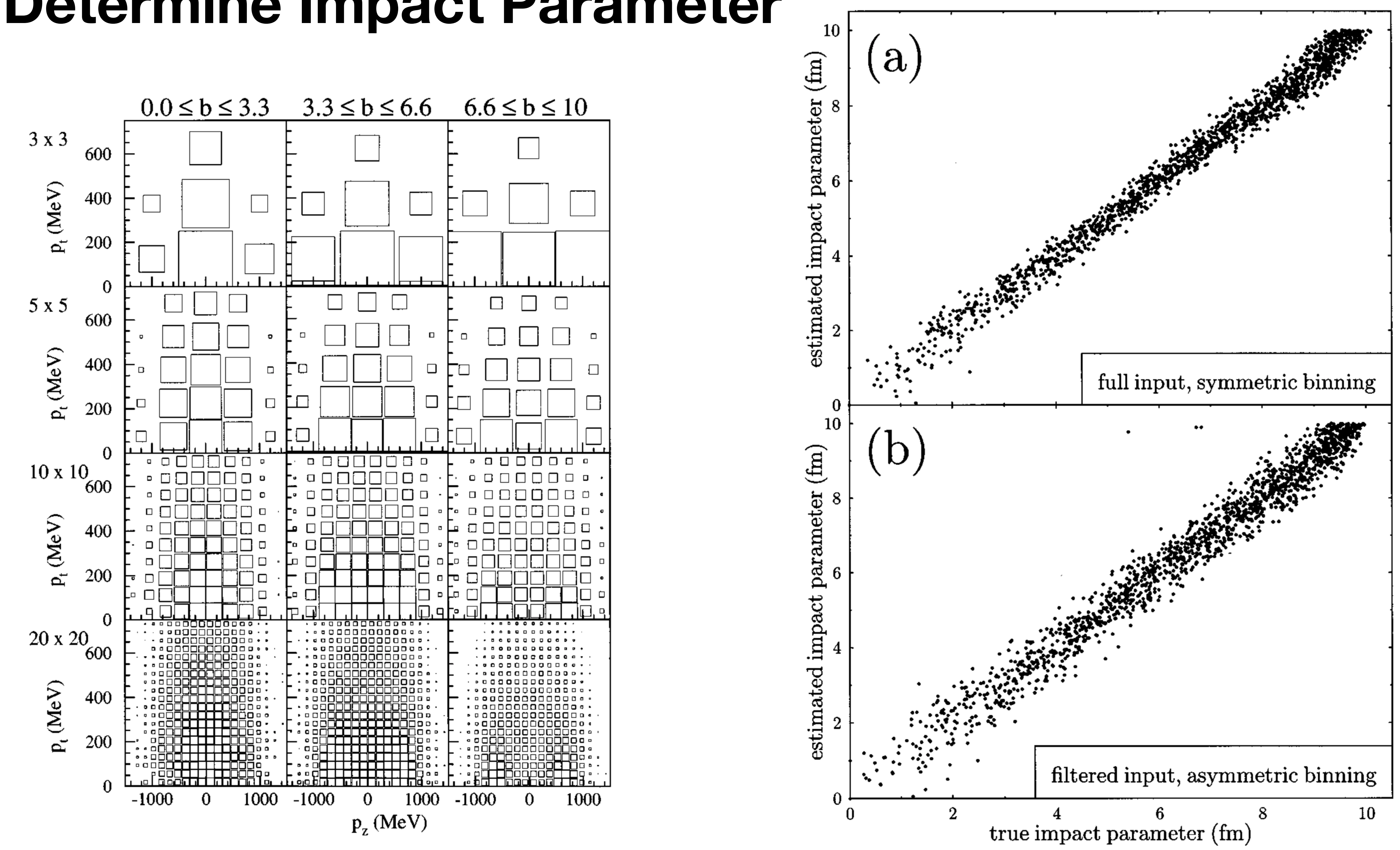
Convolutional Neural Network



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)

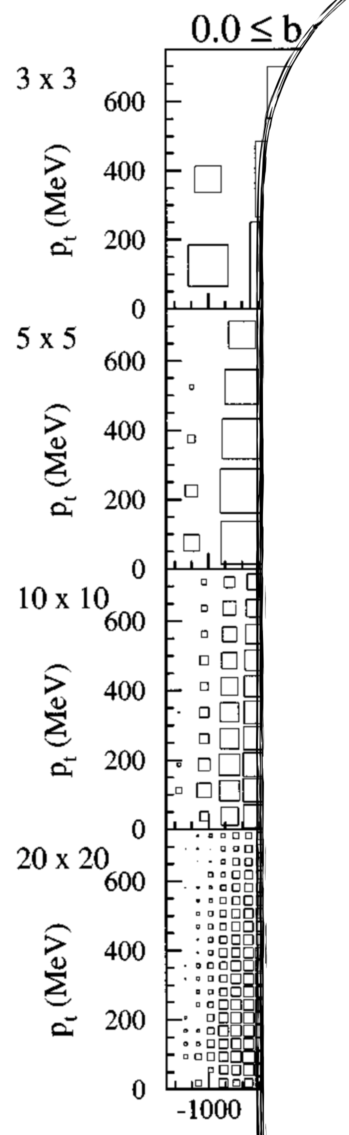
QMC data

1 hidden layer with 20 neurons

Input 5X5

Data-Driven Learning

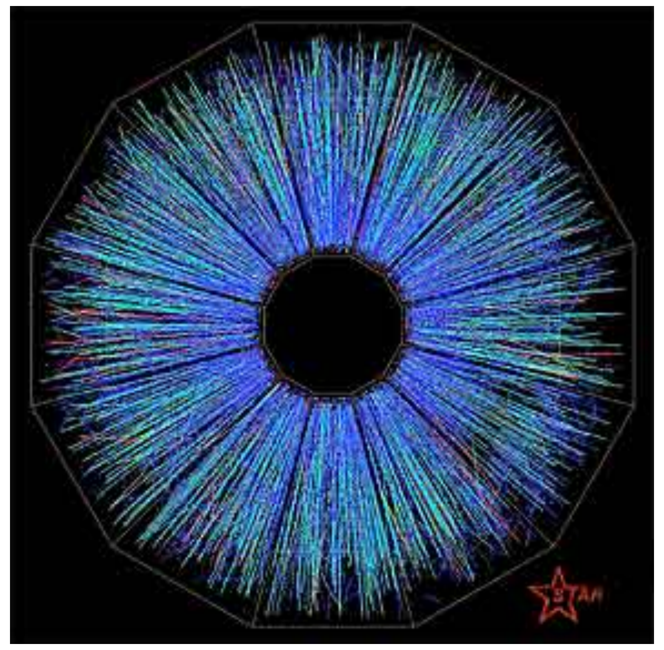
Determine Impact Parameter

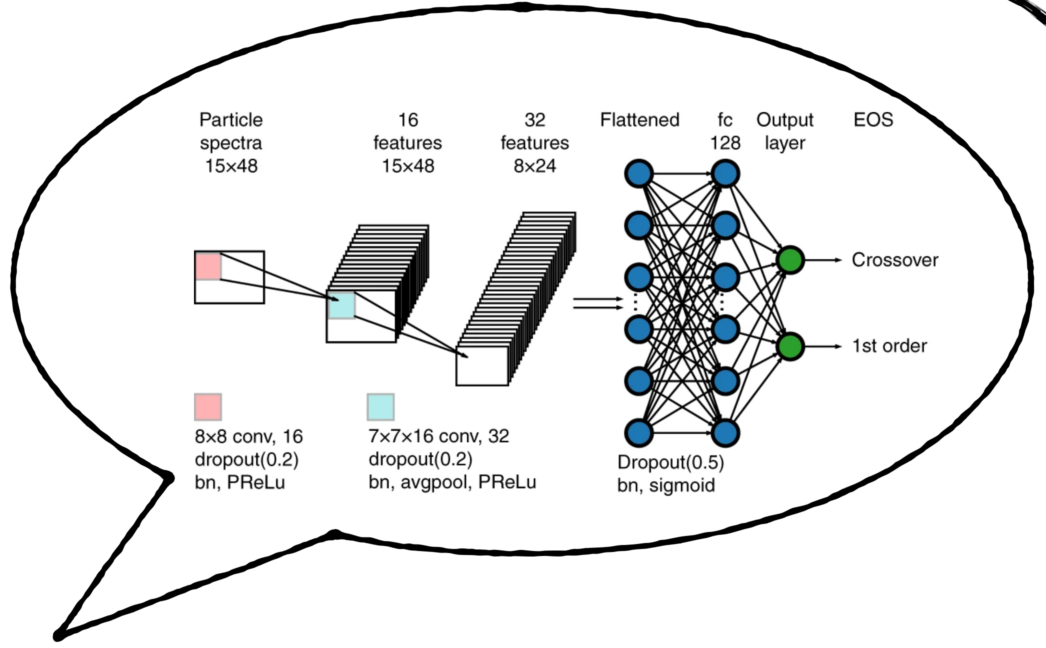


S. A

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)

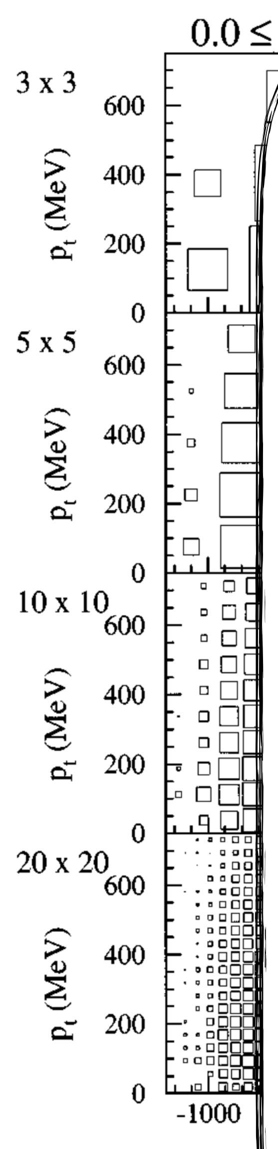
Hydro data

CNNs+DNNs

Input 15X48

Data-Driven Learning

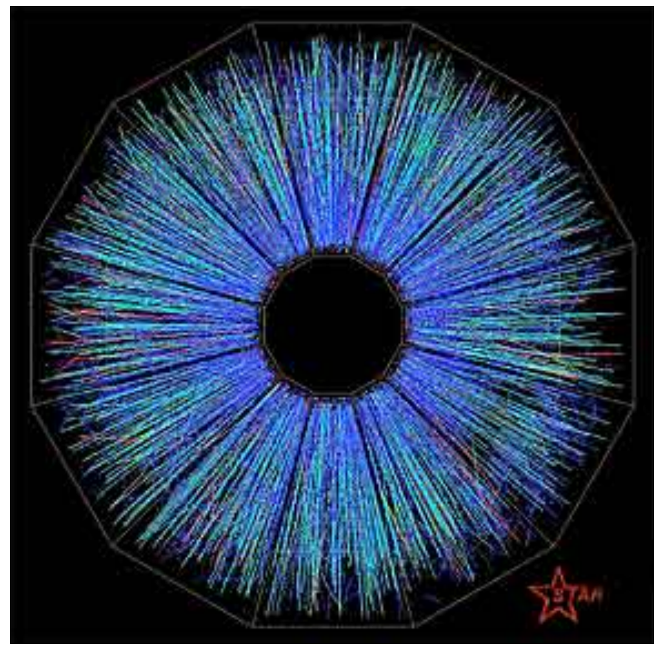
Determine Impact Parameter



S. A…

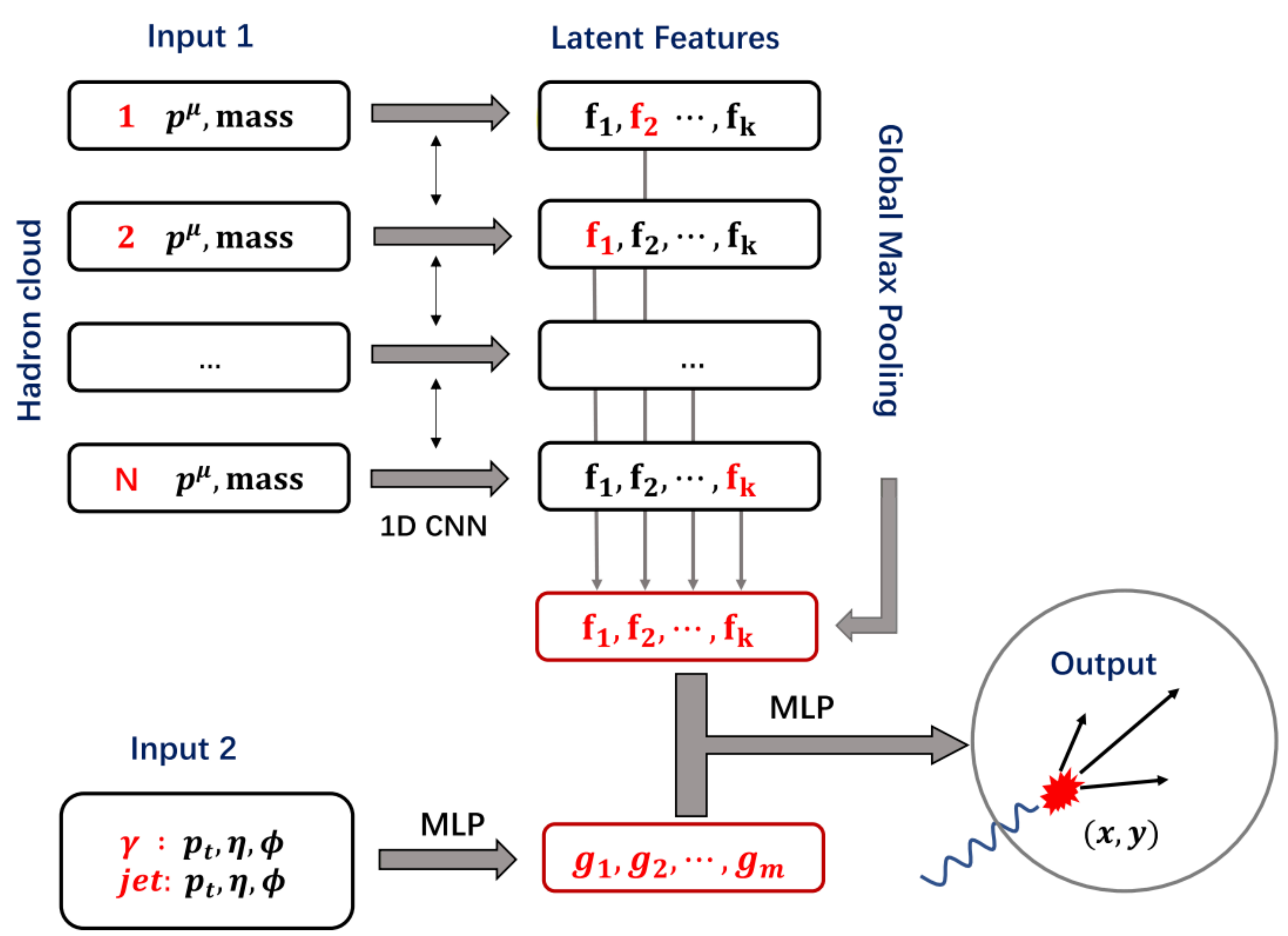
Recognizing QCD Phase Transitions

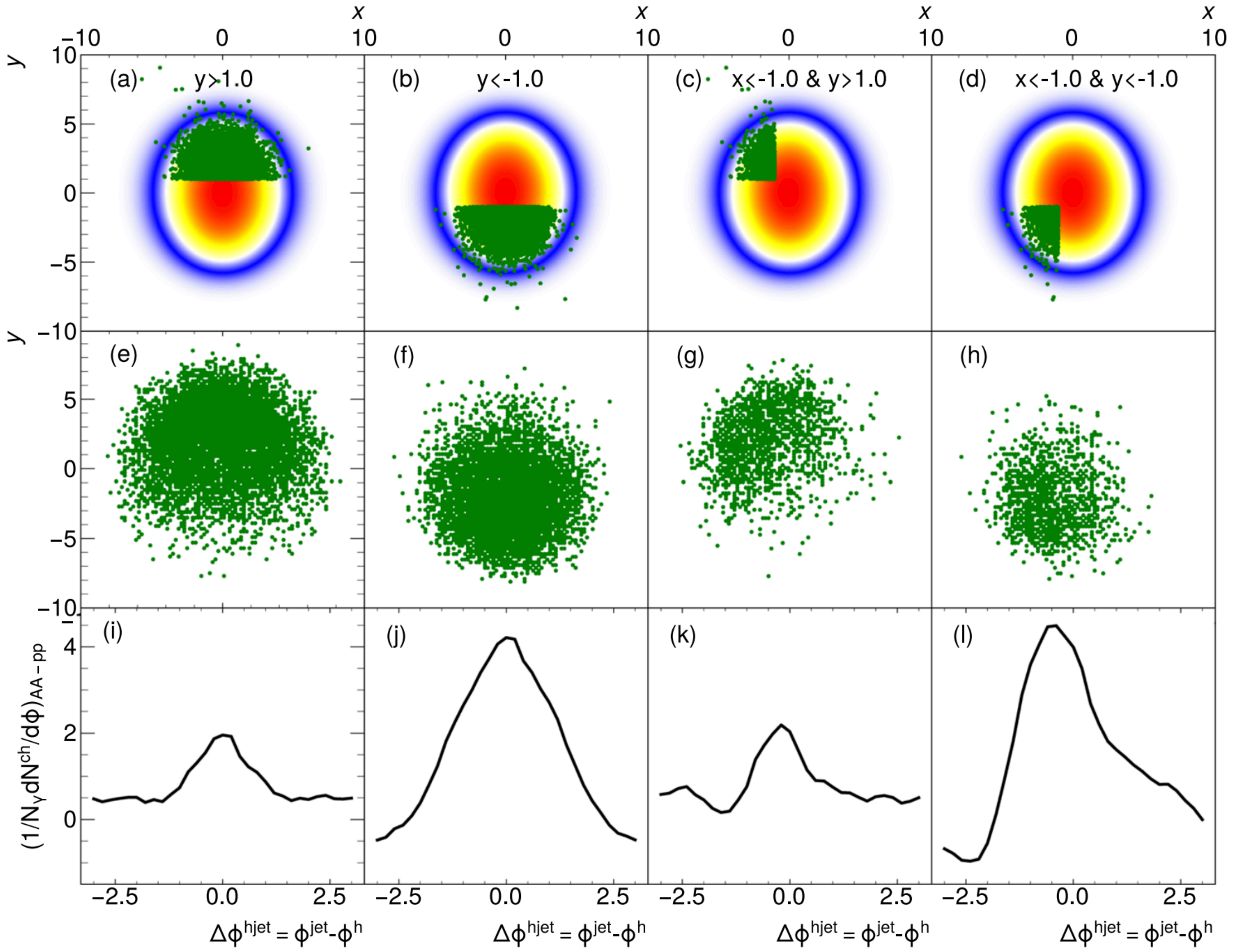
$\rho(p_T, \Phi)$



L.-G. Pang, K. Zhou, N. S…

Point Net for Jet Tomography

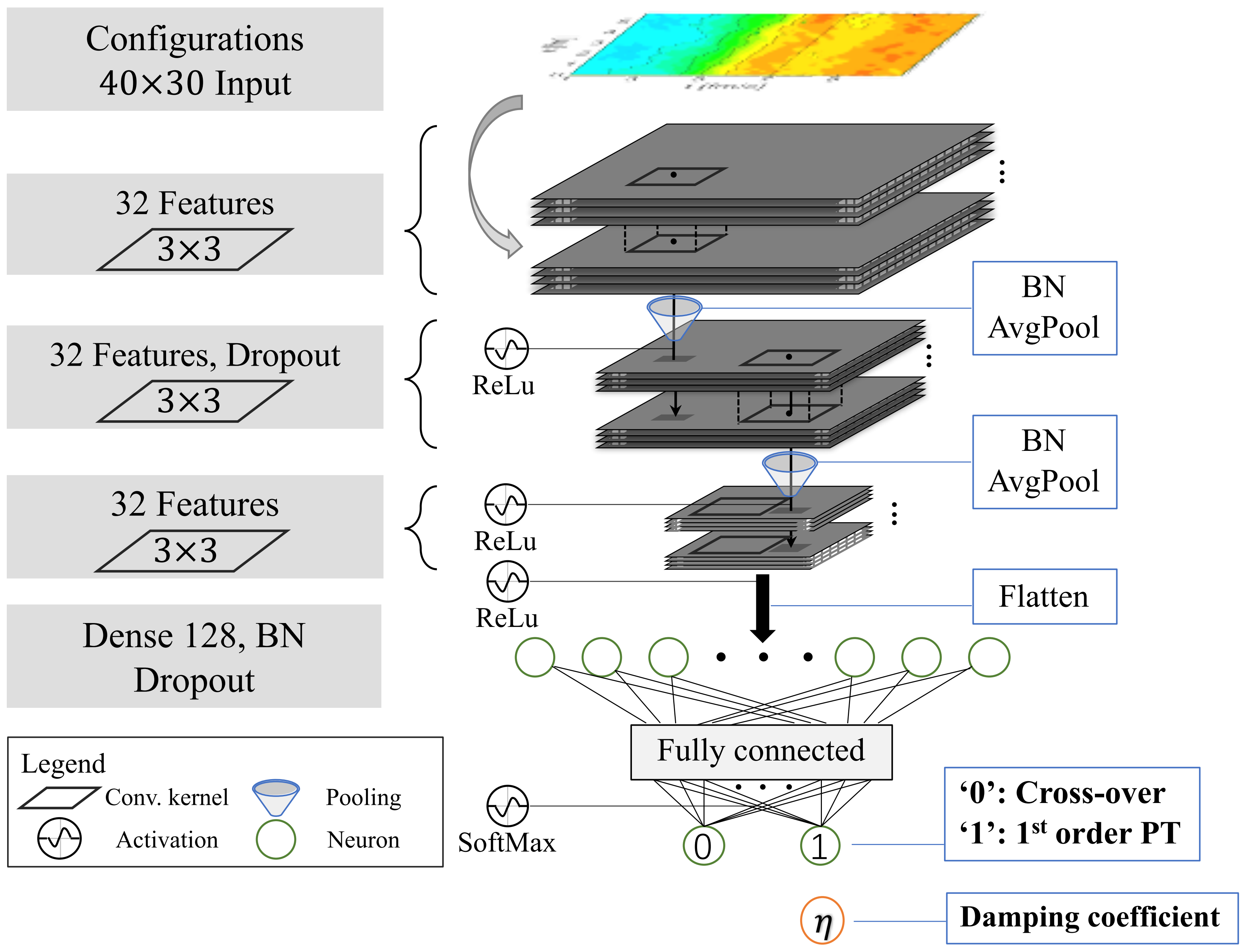




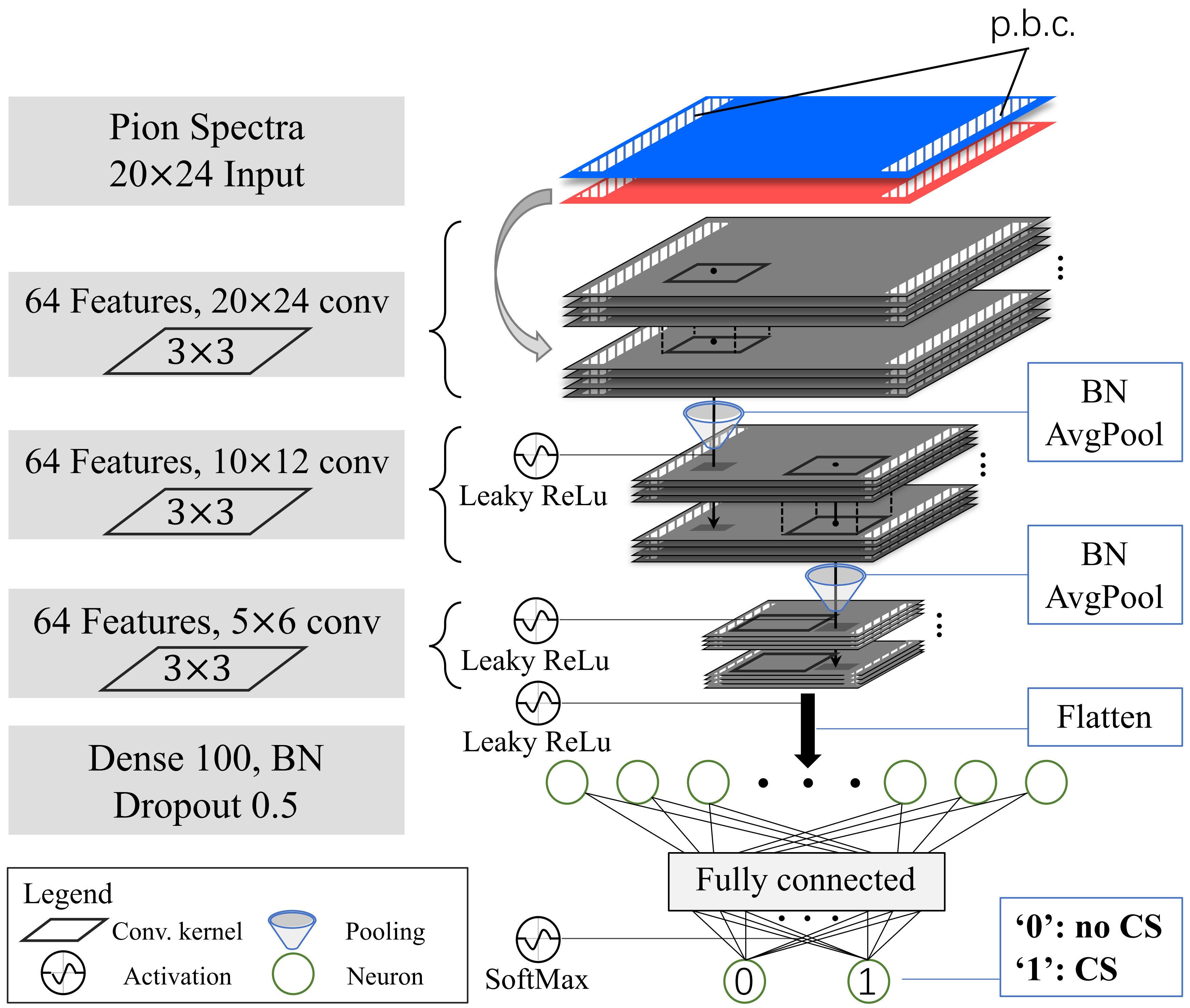
Z. Yang, Y. He, W. Chen, **W.-Y. Ke**, **L.-G. Pang**, and **X.-N. Wang**, Eur. Phys. J. C **83**, 652 (2023).

Our Current Works

Phys. Rev. D 103, 116023 **with Lijia and Kai**



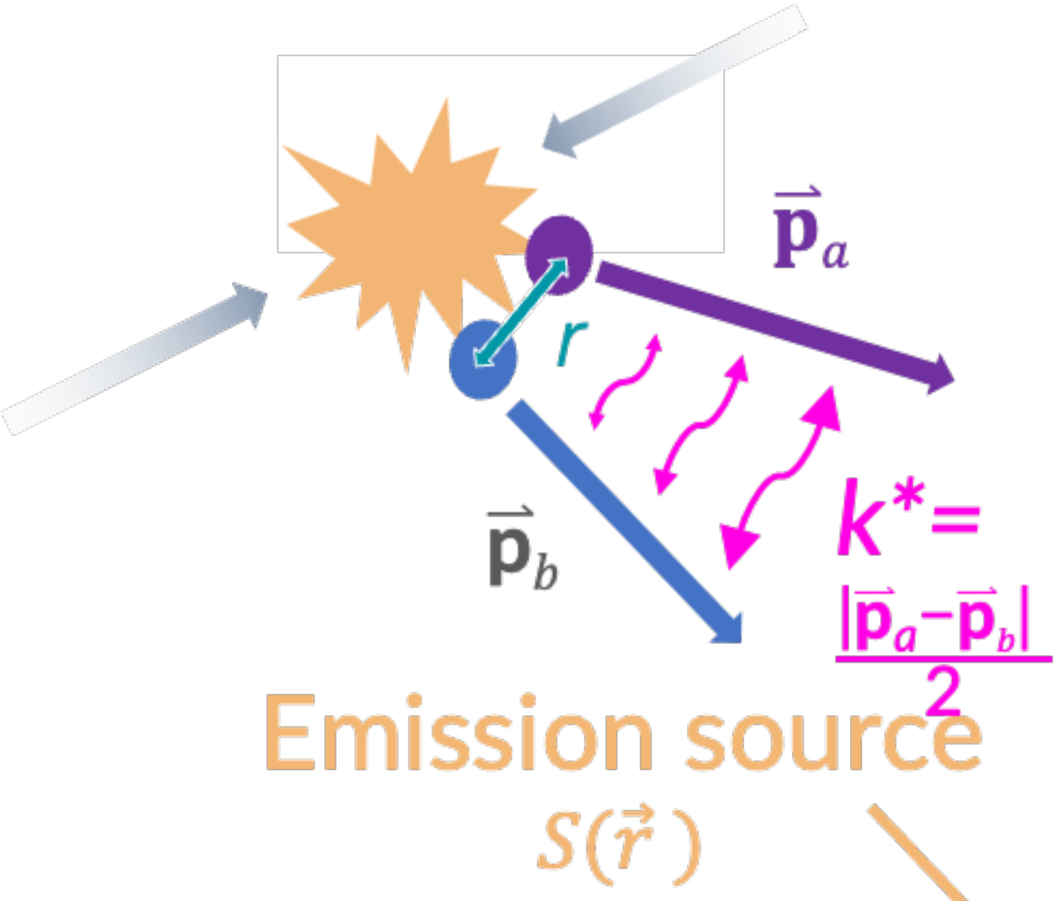
Phys. Rev. C 106, L051901(Letter) **with Yuan-Sheng, Xu-Guang and Kai**

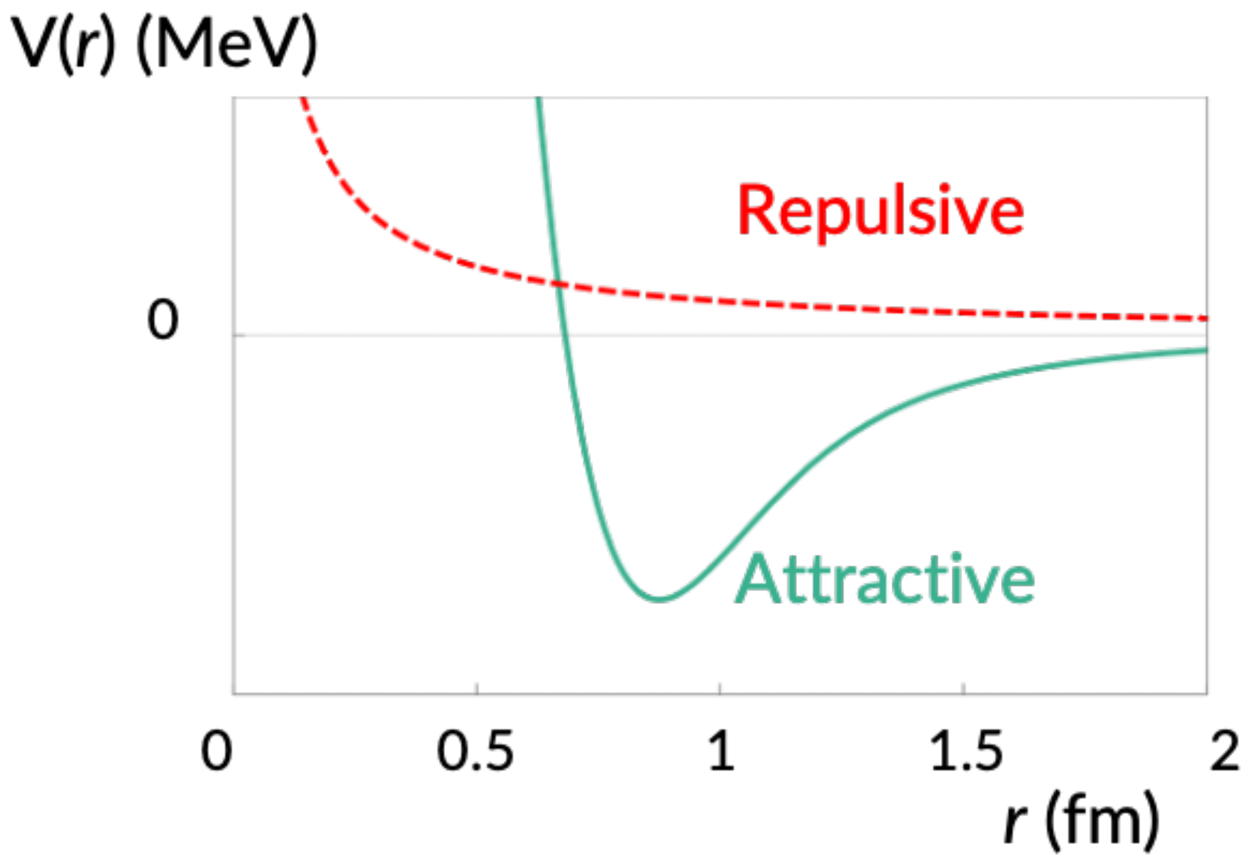


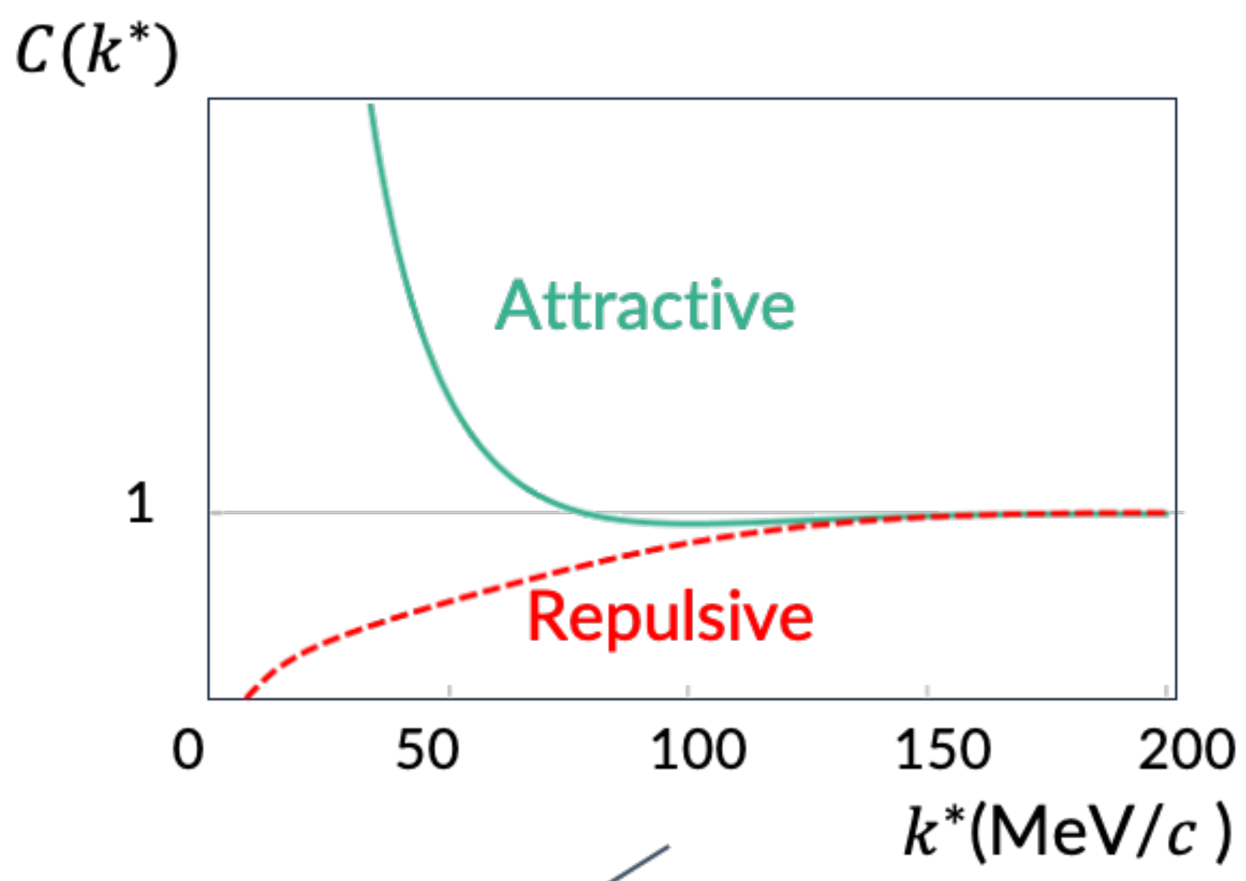
Learning phase transition orders, CME signals
from final-stage distributions

Our Current Works

in Preparation **with Jiaxing Zhao, Liang Zhang**







Two-particle wave function

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

Raffaele Del Grande | XQCD 2023

Correlations

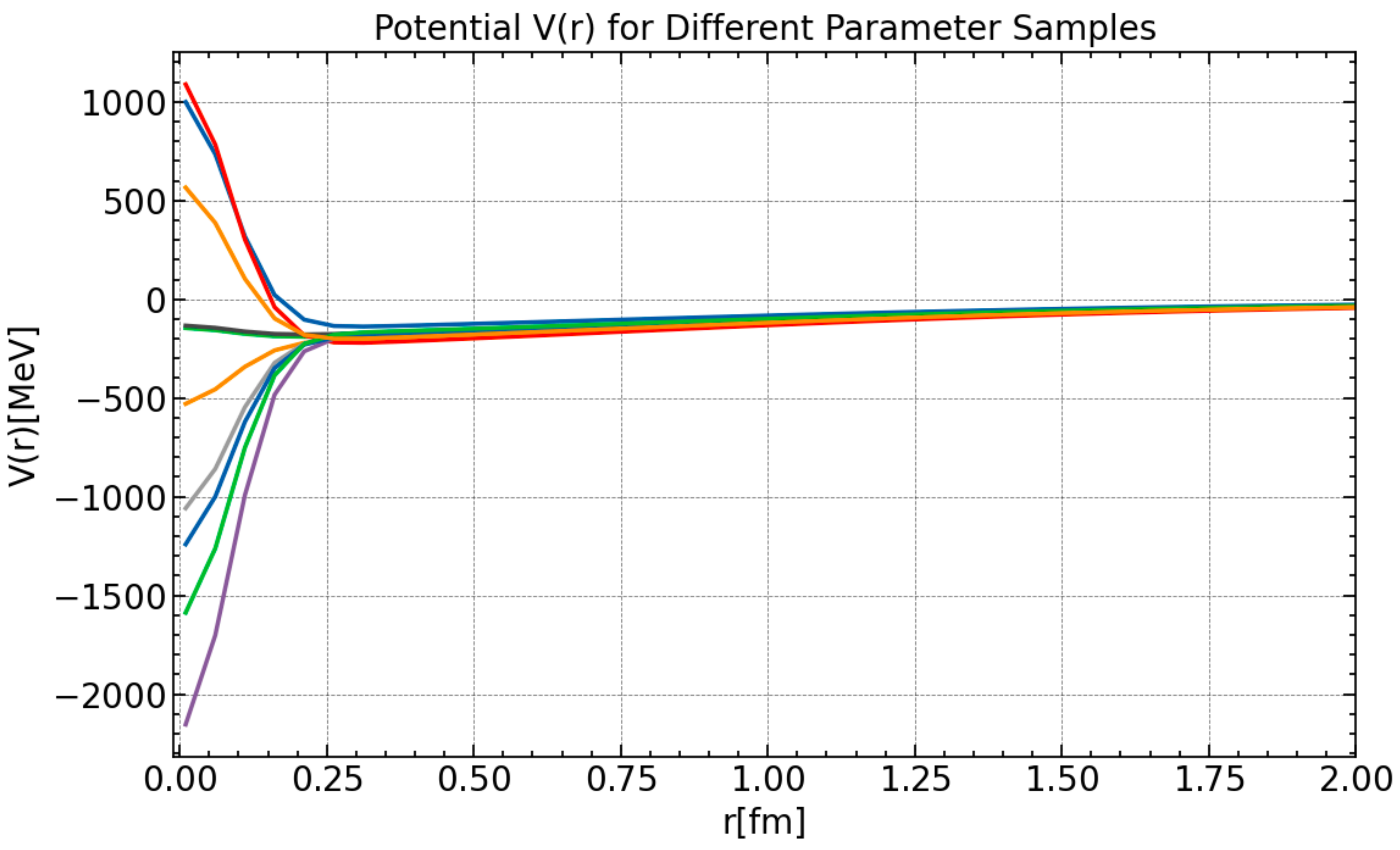
?

Potentials

Whether this inverse mapping exists?

Our Current Works

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



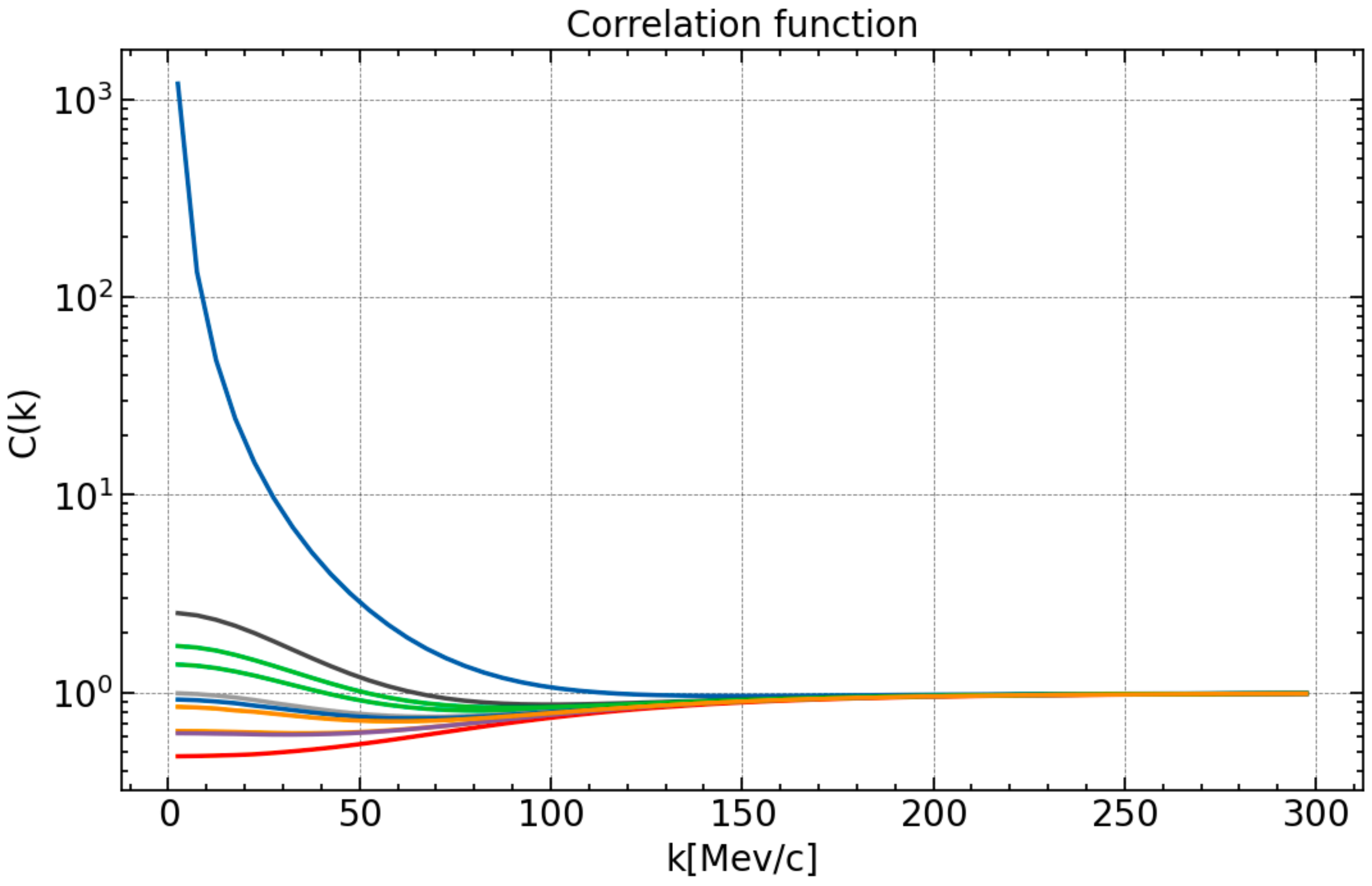
$$S(r) = (4\pi r_0^2)^{-3/2} e^{-\frac{r^2}{4r_0^2}}$$

$r_0 = 1.3$ **fm**

Schrödinger eq.

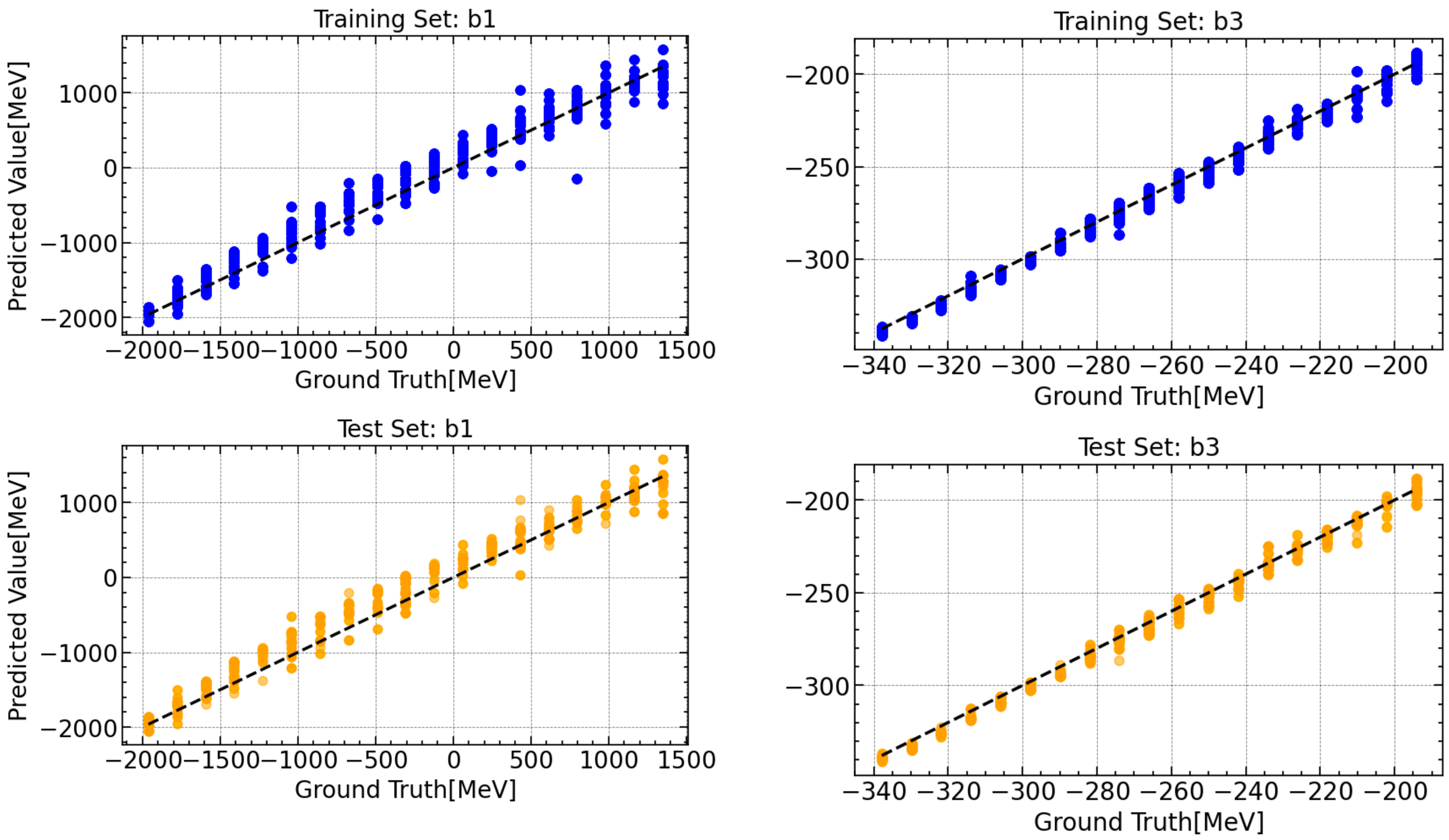
CATS Framework: D. Mihaylov et al., Eur. Phys. J. C78 (2018) 394

in Preparation **with Jiaxing Zhao, Liang Zhang**



60 k-points, 10000 correlations

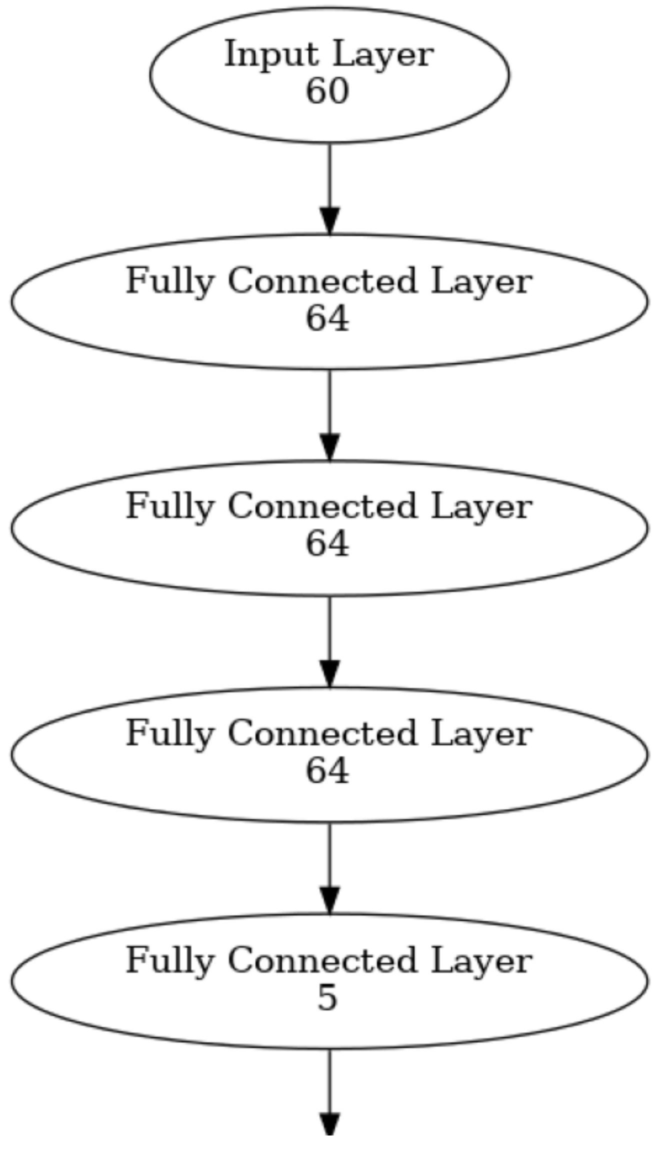
Our Current Works



in Preparation

with Jiaxing Zhao, Liang Zhang

60 points



b_1, b_3

R-squared	b1	b3
Training	0.96	0.99
Testing	0.96	0.99

Neural networks for Femtoscopy

$b_2 = 73.9, b_4 = 0.78, n_\pi = 2$

Physics-Driven Learning

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

Forward process

Physics

Model Parameters/ Properties/States

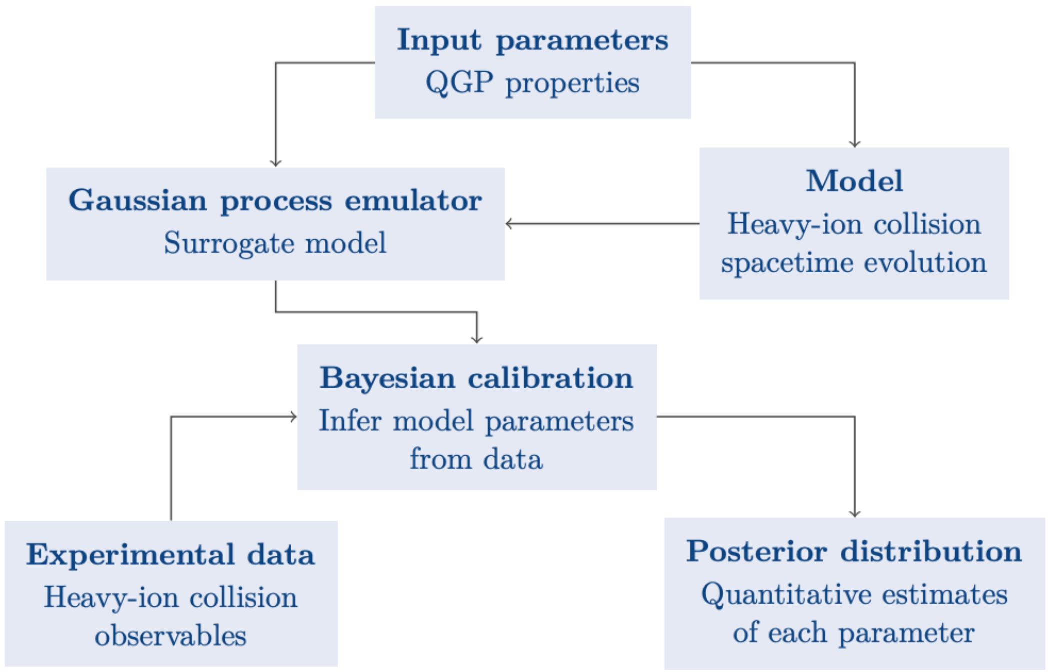
Data

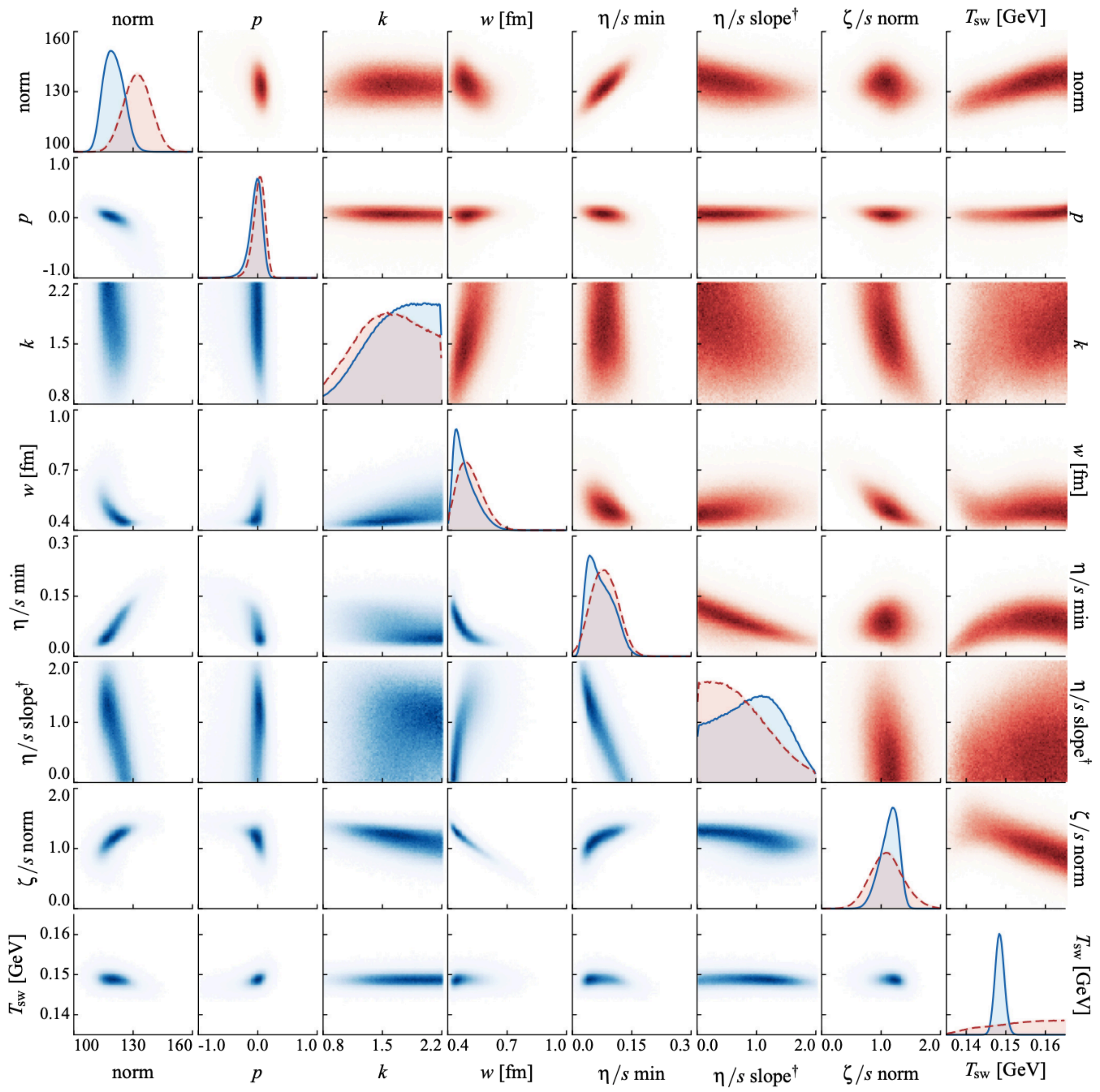
Observations

Inference

Physics-Driven Learning

Inferring Dynamical Information

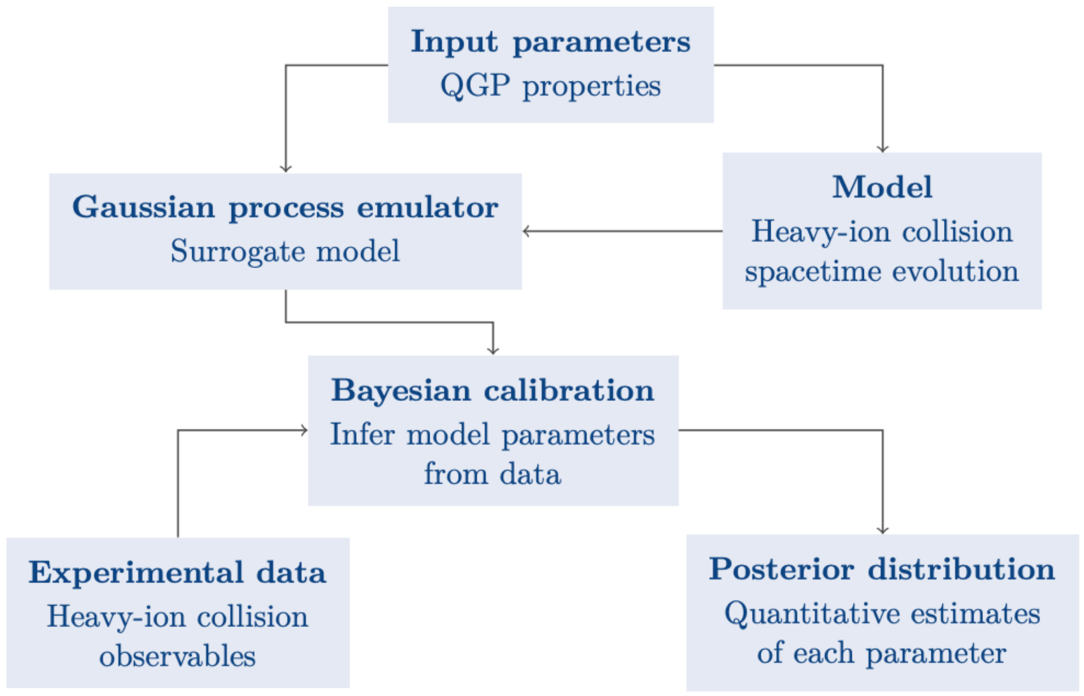




J. E. Bernhard, etc.(Duke Group), arXiv:1804.06469 and Nat Phy **15**, 1113 (2019).

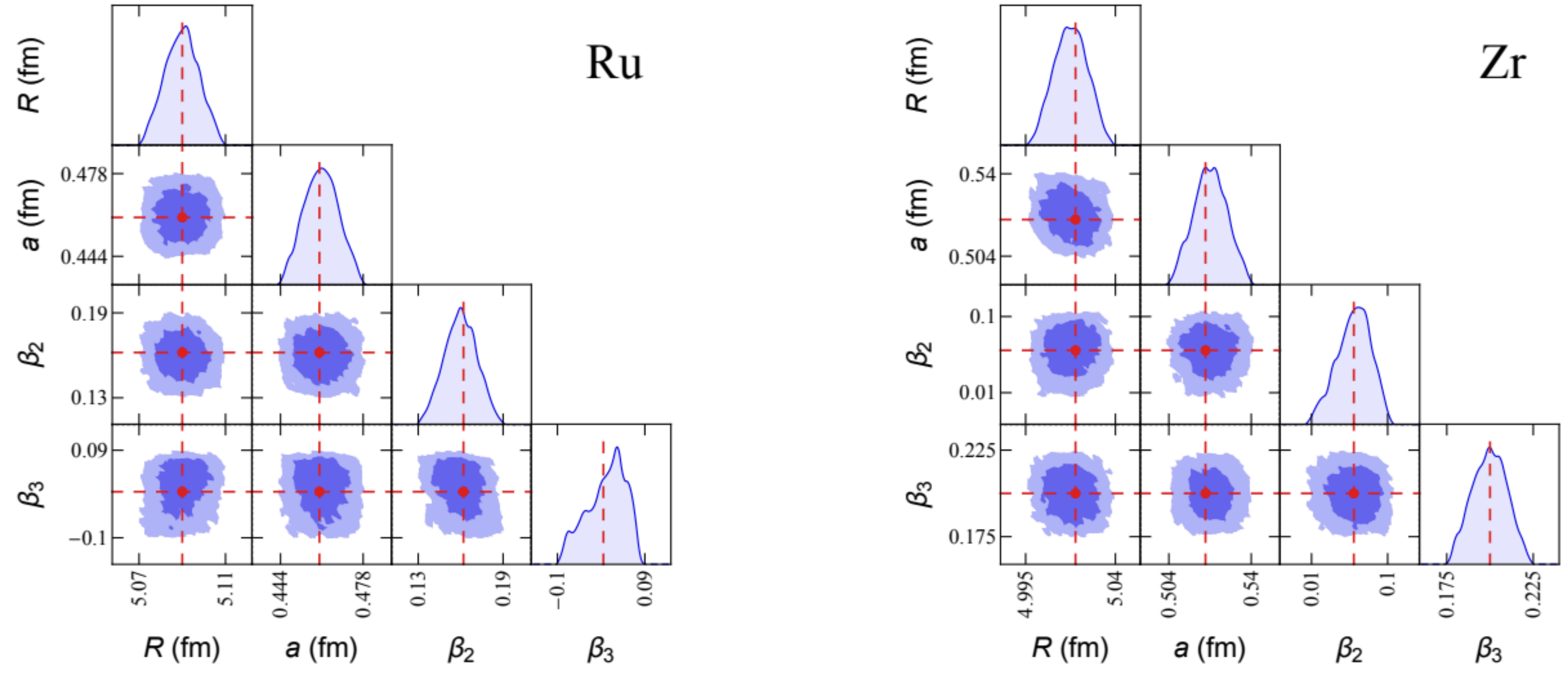
Physics-Driven Learning

Inferring Dynamical Information



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

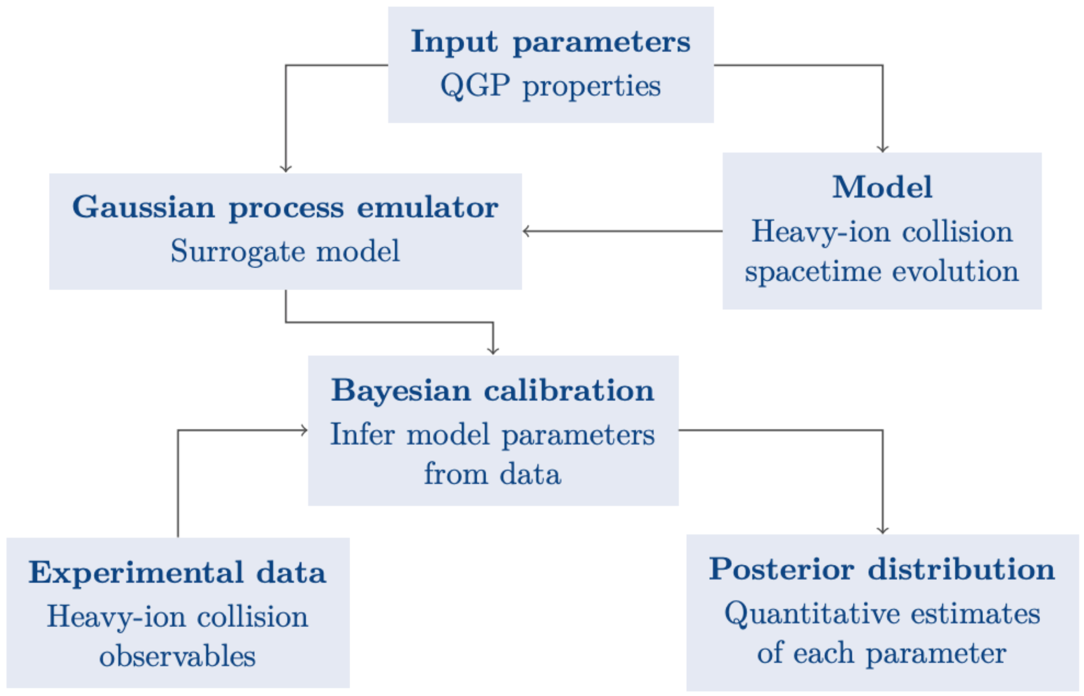
Determing Nulcear Struture



Y.-L. Cheng, **S. Shi, Y.-G. Ma**, H. Stöcker, and **K. Zhou**, Phys. Rev. C **107**, 064909 (2023)

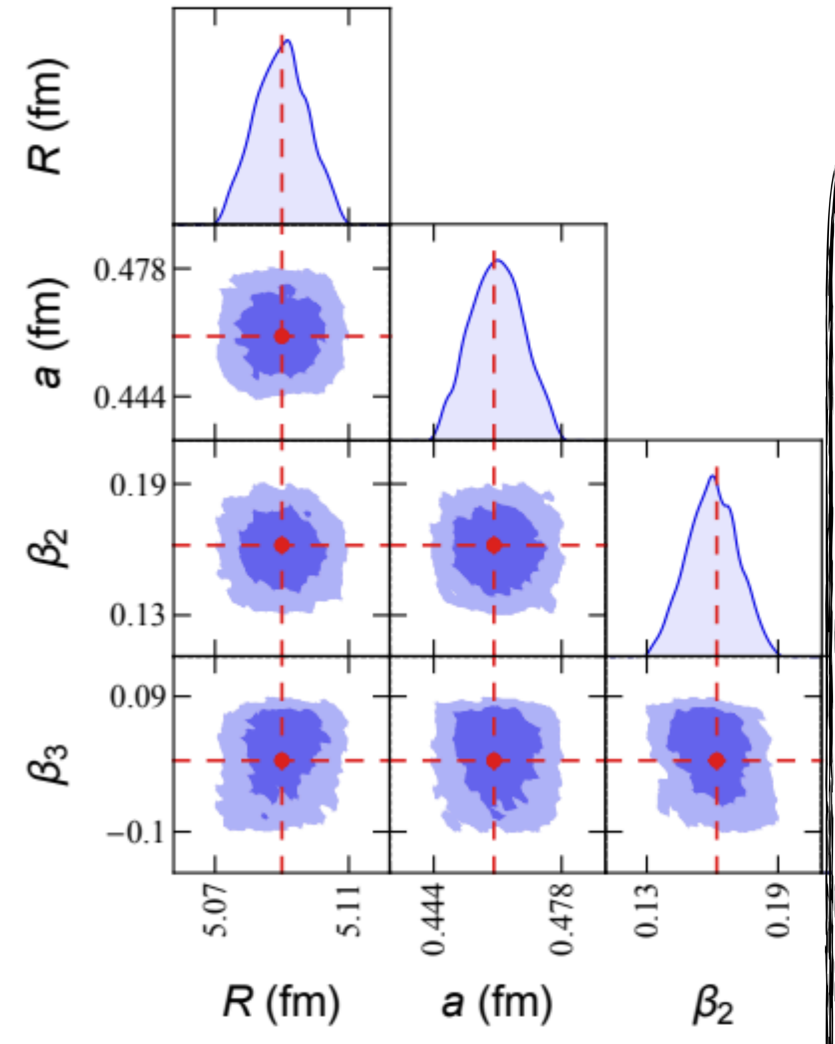
Physics-Driven Learning

Inferring Dynamical Information



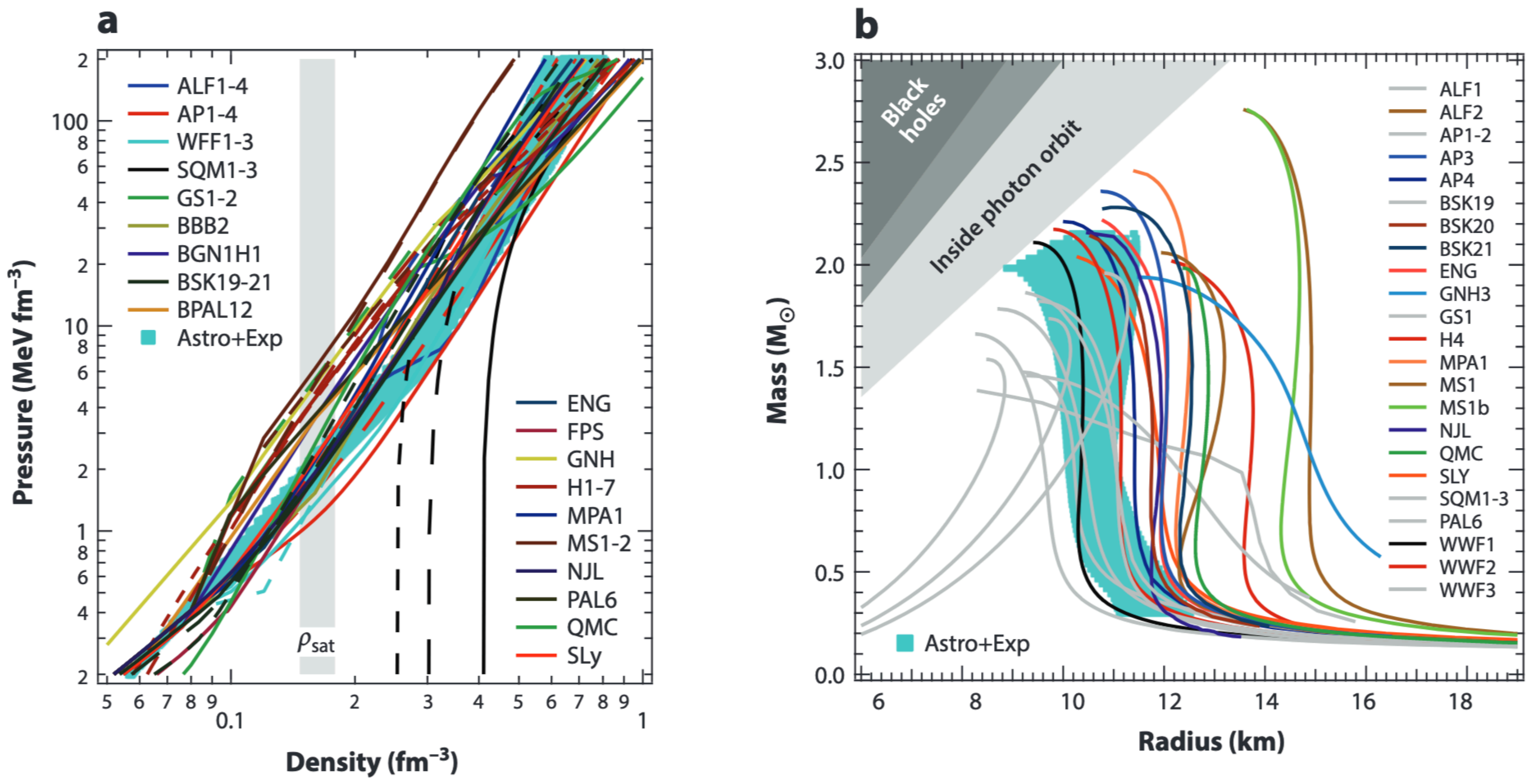
J. E. Bernhard, etc.(Duke Group), arXiv:1804.06469

Determing Nulcear Struture



Y.-L. Cheng, S. Shi, Y.

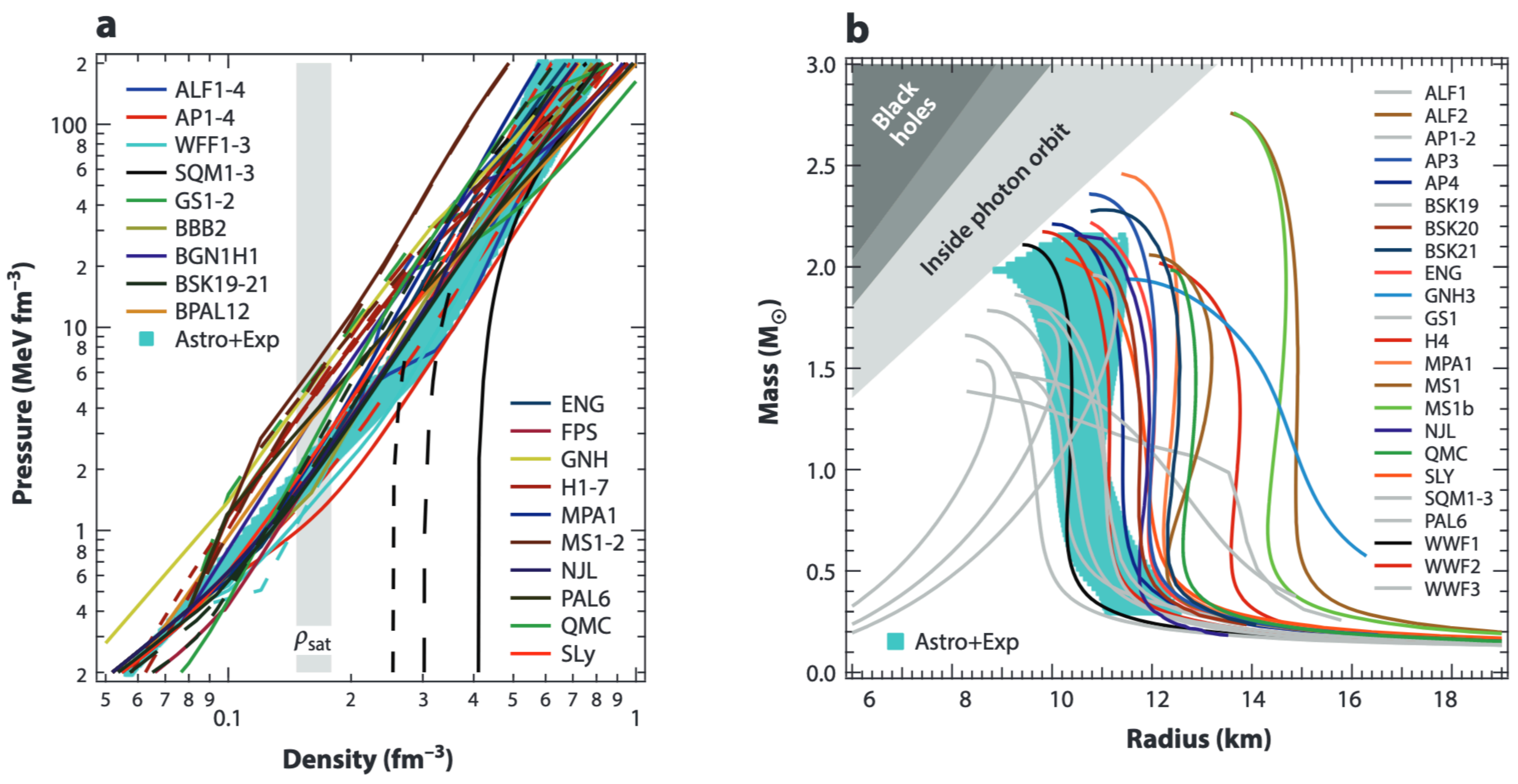
Building Nuclear Matter EoS



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

Physics-Driven Learning

Building Nuclear Matter EoS



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

Physics Parameters are Finite

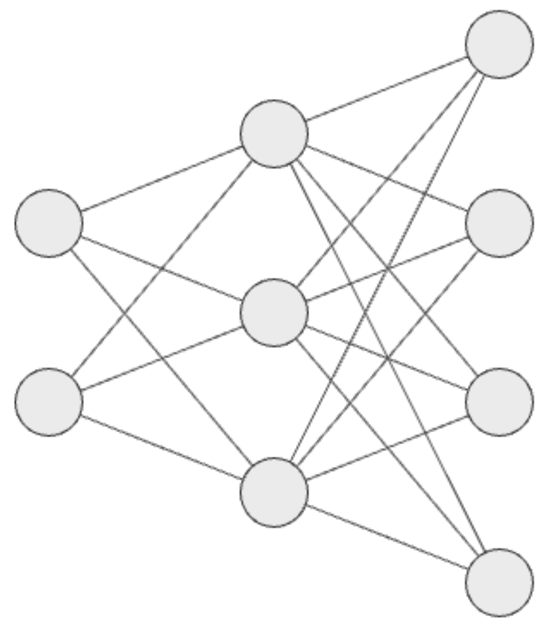
EoS, Wave-Function, Potential, …

Inference is Easy-To-Compute

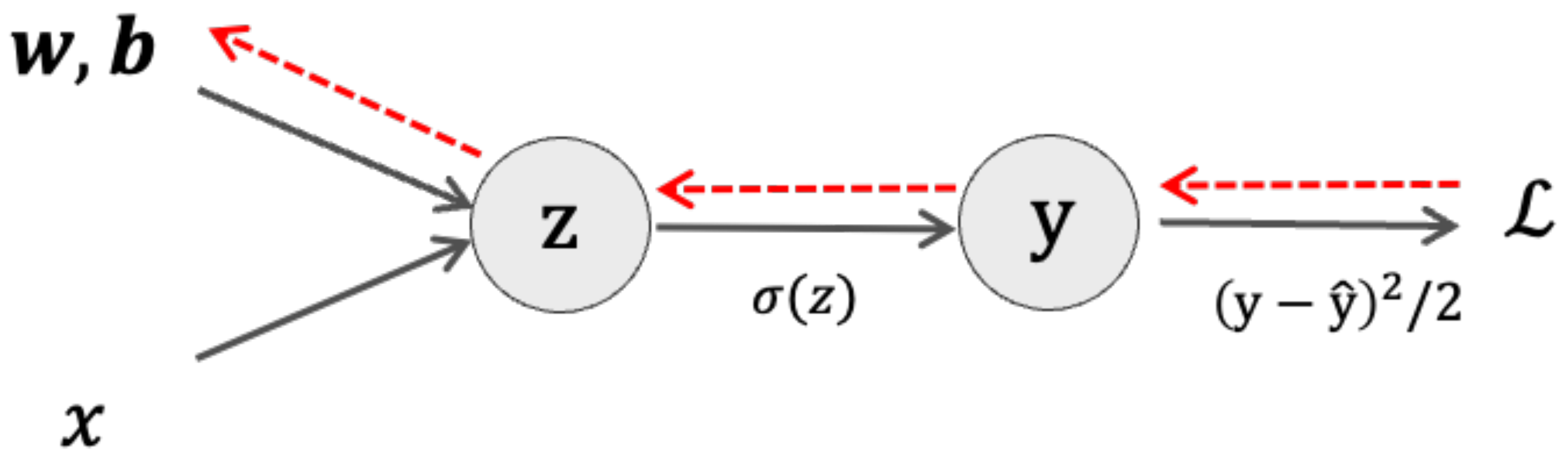
ODEs, PDEs, Simulations, …

Physics-Driven Deep Learning

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



$f_\theta \underset{\text{BP}}{\overset{\text{Phys}}{\rightleftarrows}} \quad \tilde{O} \qquad \mathcal{L} = (O - \tilde{O})^2$



Deep Neural Network represented Physics, f_θ

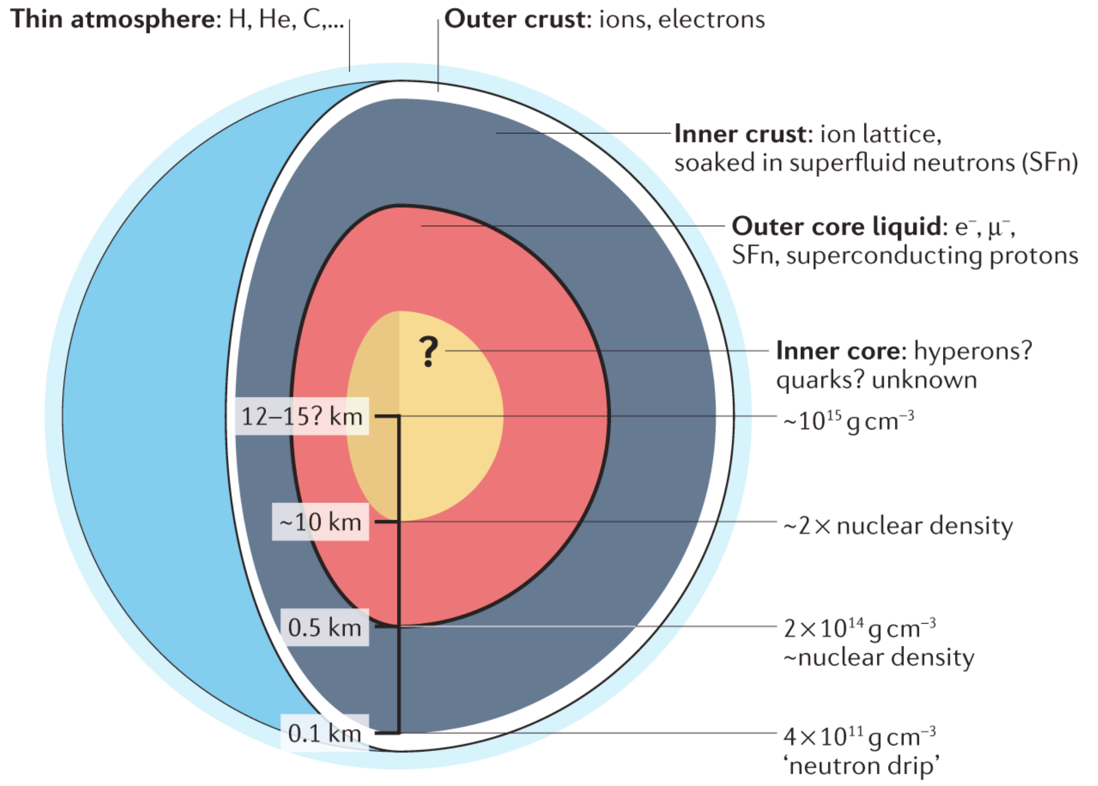
Flexible Representation

Back-Propagation

Easy-To-Compute on GPUs

Physics-Driven Deep Learning

1. Building Neutron Star EoS

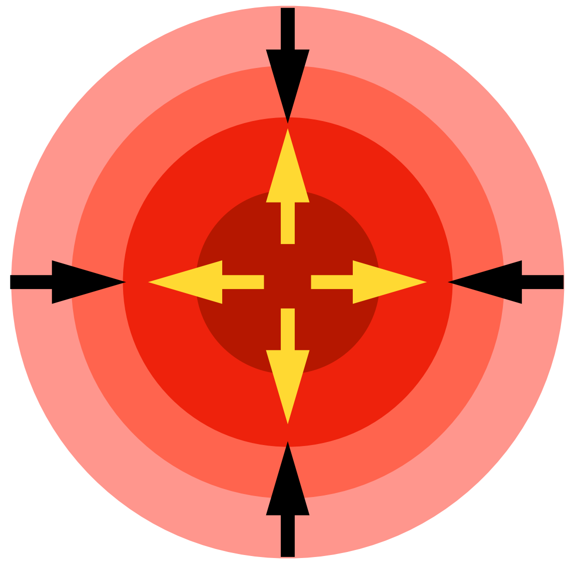


Nat. Rev. Phys. 4, 237–246 (2022)

Tolman–Oppenheimer–Volkoff equations

$$\begin{cases} \dfrac{dP}{dr} = -G\dfrac{m(r)\varepsilon(r)}{r^2}\left(1+\dfrac{P(r)}{\varepsilon(r)}\right)\left(1+\dfrac{4\pi r^3 P(r)}{m(r)}\right)\left(1-\dfrac{2Gm(r)}{r}\right)^{-1} \\ \dfrac{dm(r)}{dr} = 4\pi r^2 \epsilon(r) \end{cases}$$

Hydrostatic condition in each shell (dr)

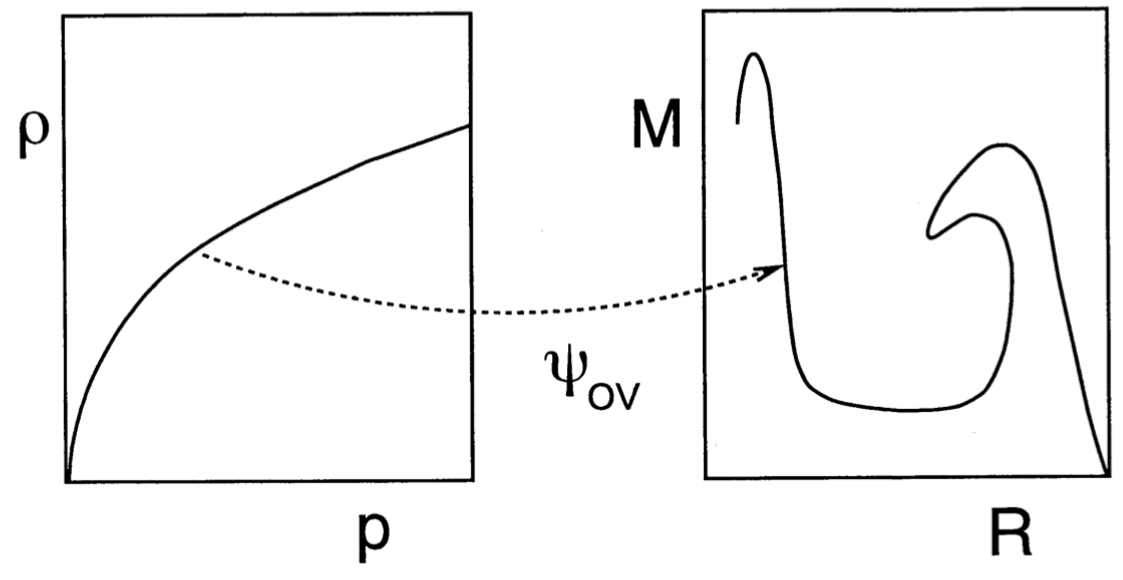


EoS $P(\varepsilon) = 0$

Core $r = 0, \varepsilon(r=0) = \varepsilon_c, P(r=0) = P(\varepsilon_c)$

Surface $r = R, \varepsilon(r=R) \simeq 0, M = \int 4\pi r^2 \varepsilon(r) dr$

M, R



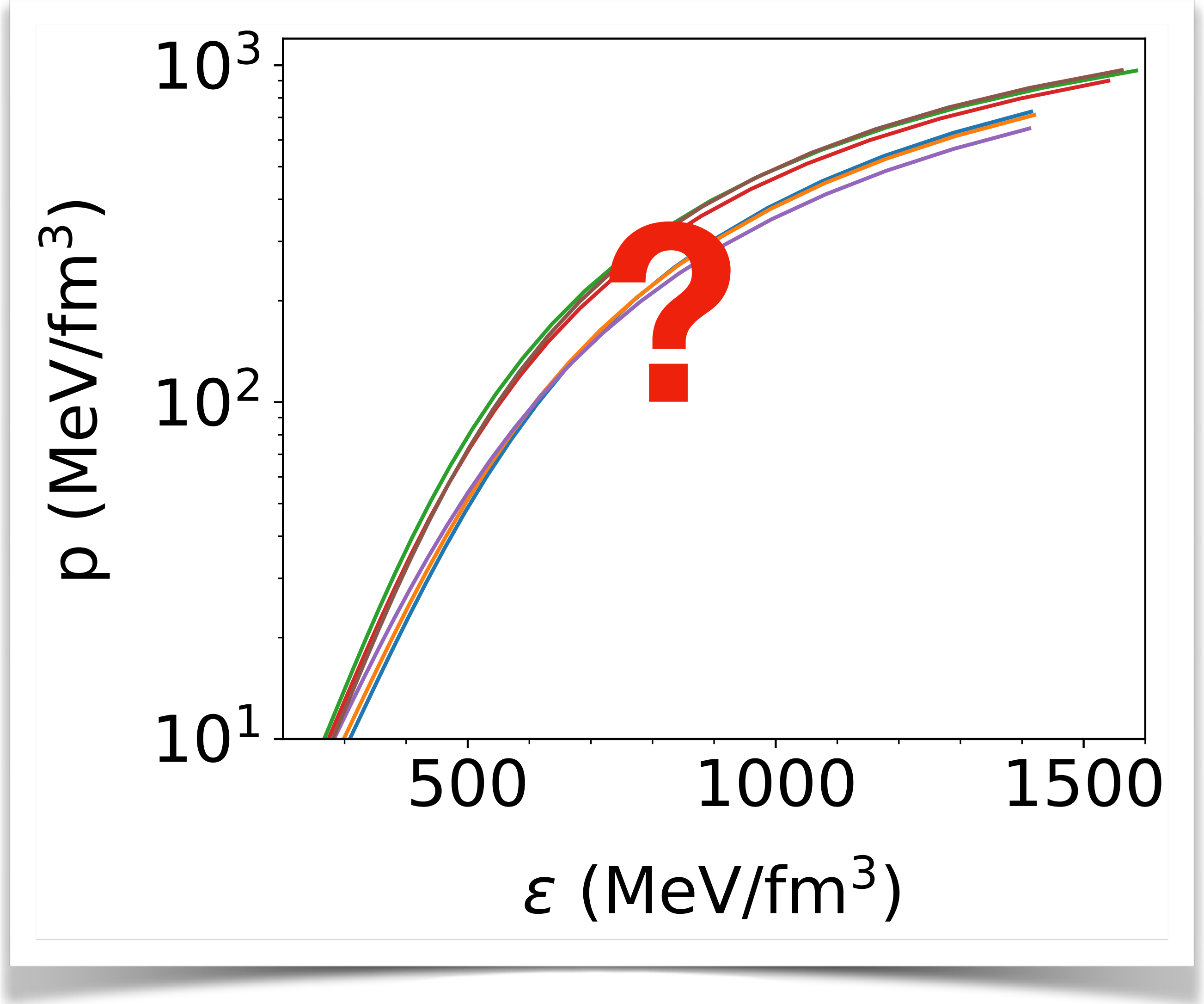
L. Lindblom, *A.J., 398, 569 (1992).*
If the **whole M(R)** is known,it's well-defined problem.

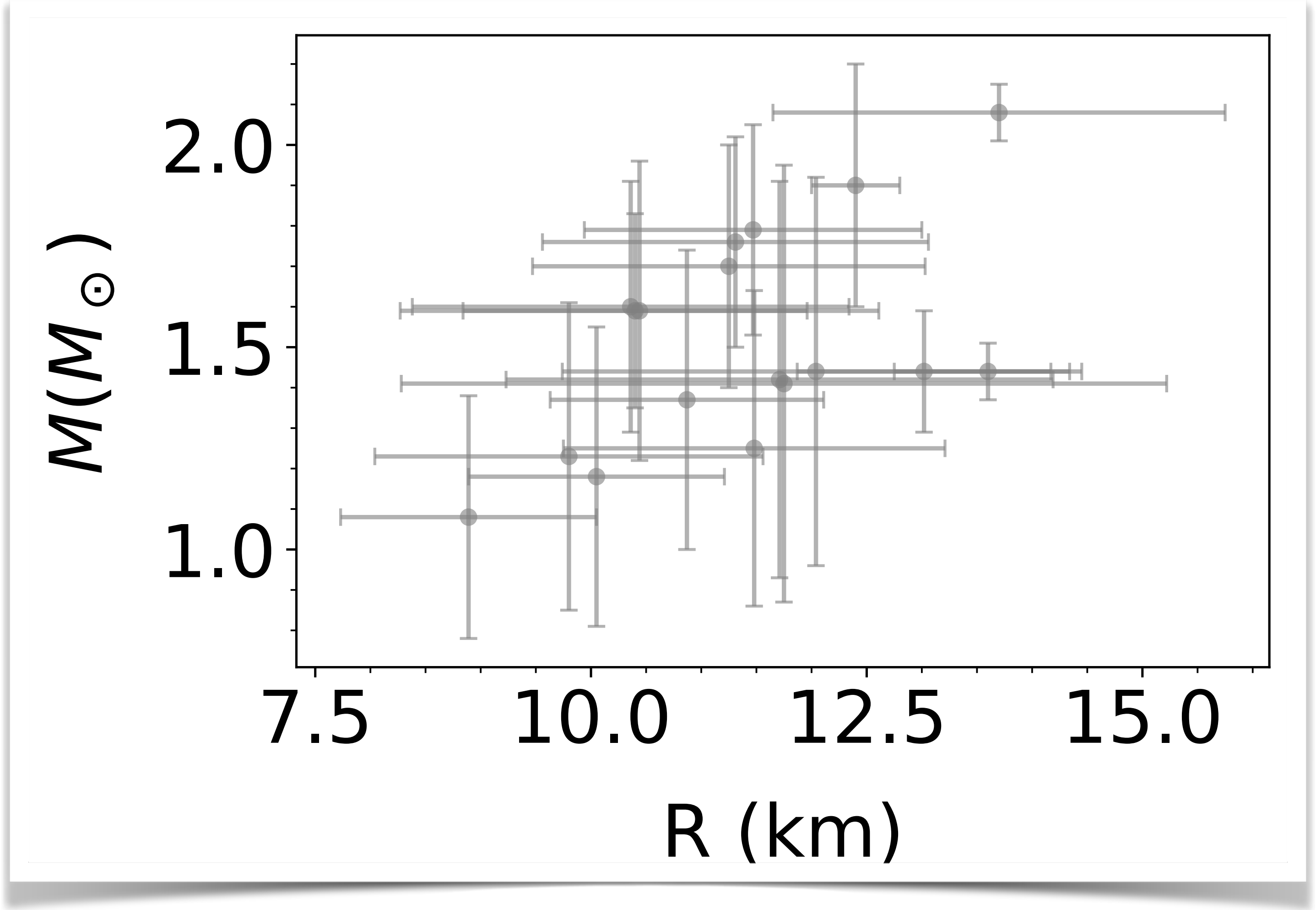
Pressure ➡ ⬅ Gravity

Physics-Driven Deep Learning

1. Building Neutron Star EoS

Tolman–Oppenheimer–Volkoff equations

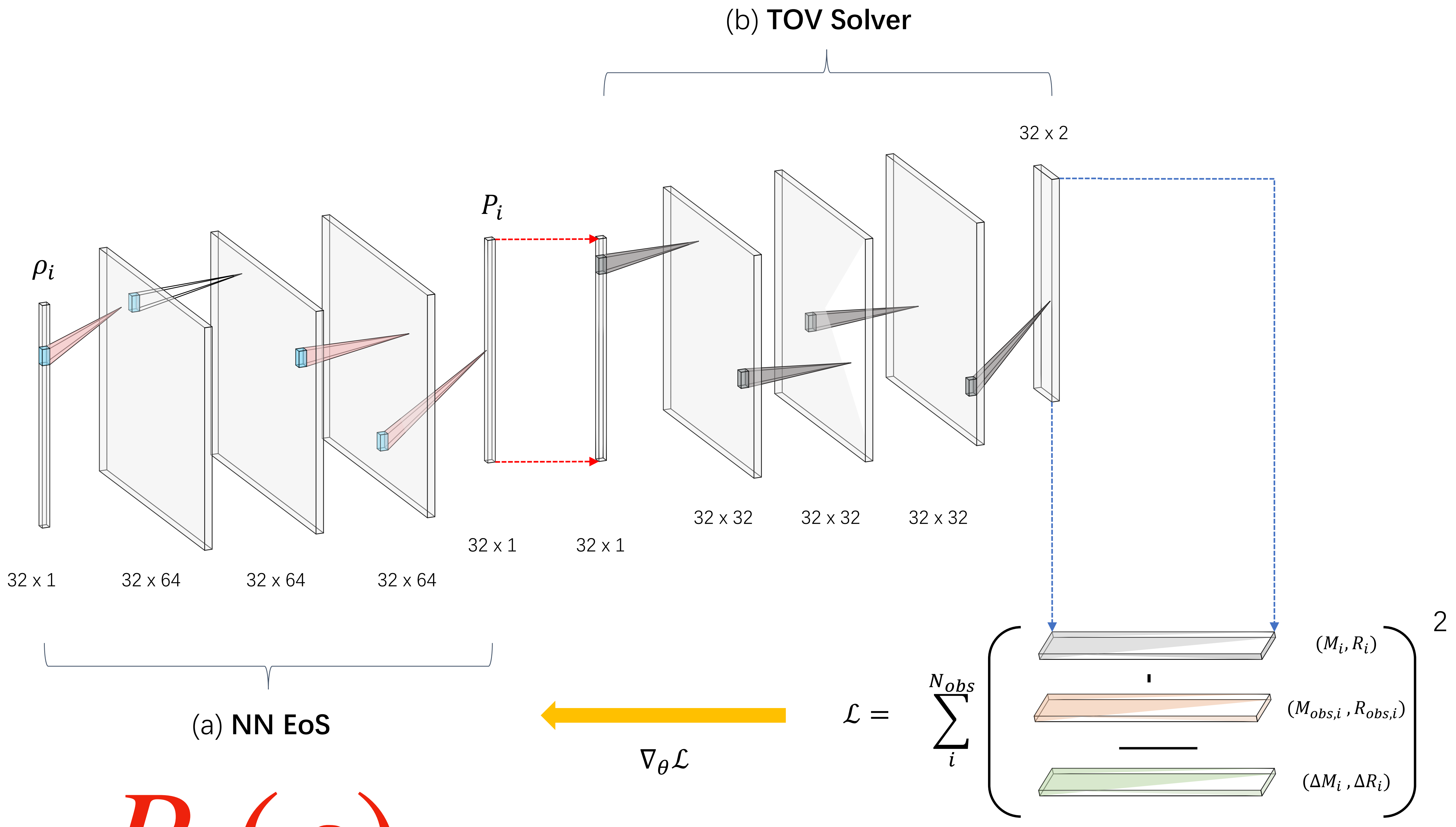




Physics-Driven Deep Learning

1. Building Neutron Star EoS

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



$$P_\theta(\rho)$$

$$\mathscr{L} = \chi^2 = \sum_{i=1}^{N_{\rm obs}} \frac{\left(M_i - M_{{\rm obs},i}\right)^2}{\Delta M_i^2} + \frac{\left(R_i - R_{{\rm obs},i}\right)^2}{\Delta R_i^2}$$

Loss function

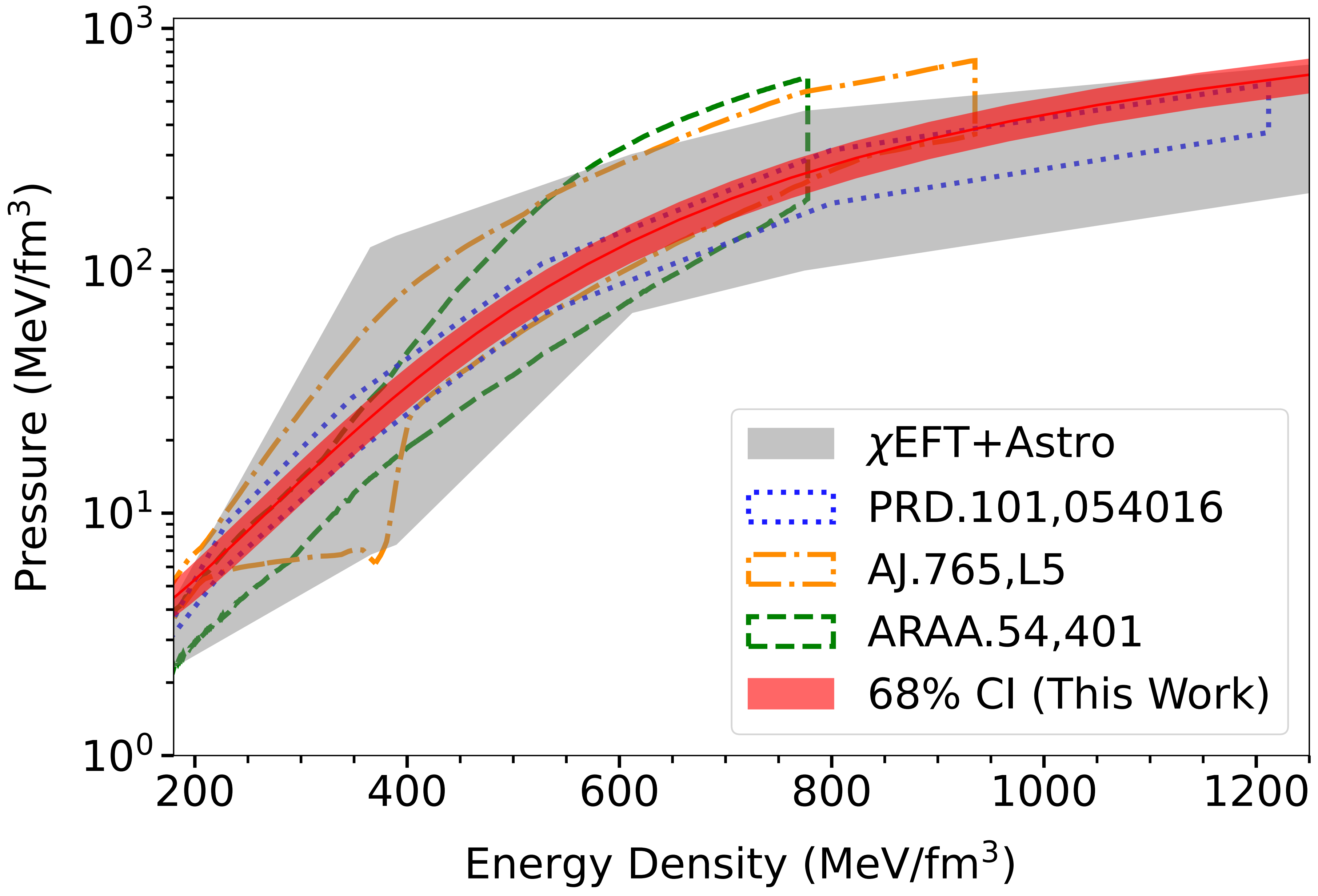
- (M_i, R_i) : predicted Mass-Radius
- $(M_{obs,i}, R_{obs,i})$: observed Mass-Radius
- $(\Delta M_i, \Delta R_i)$: uncertainties

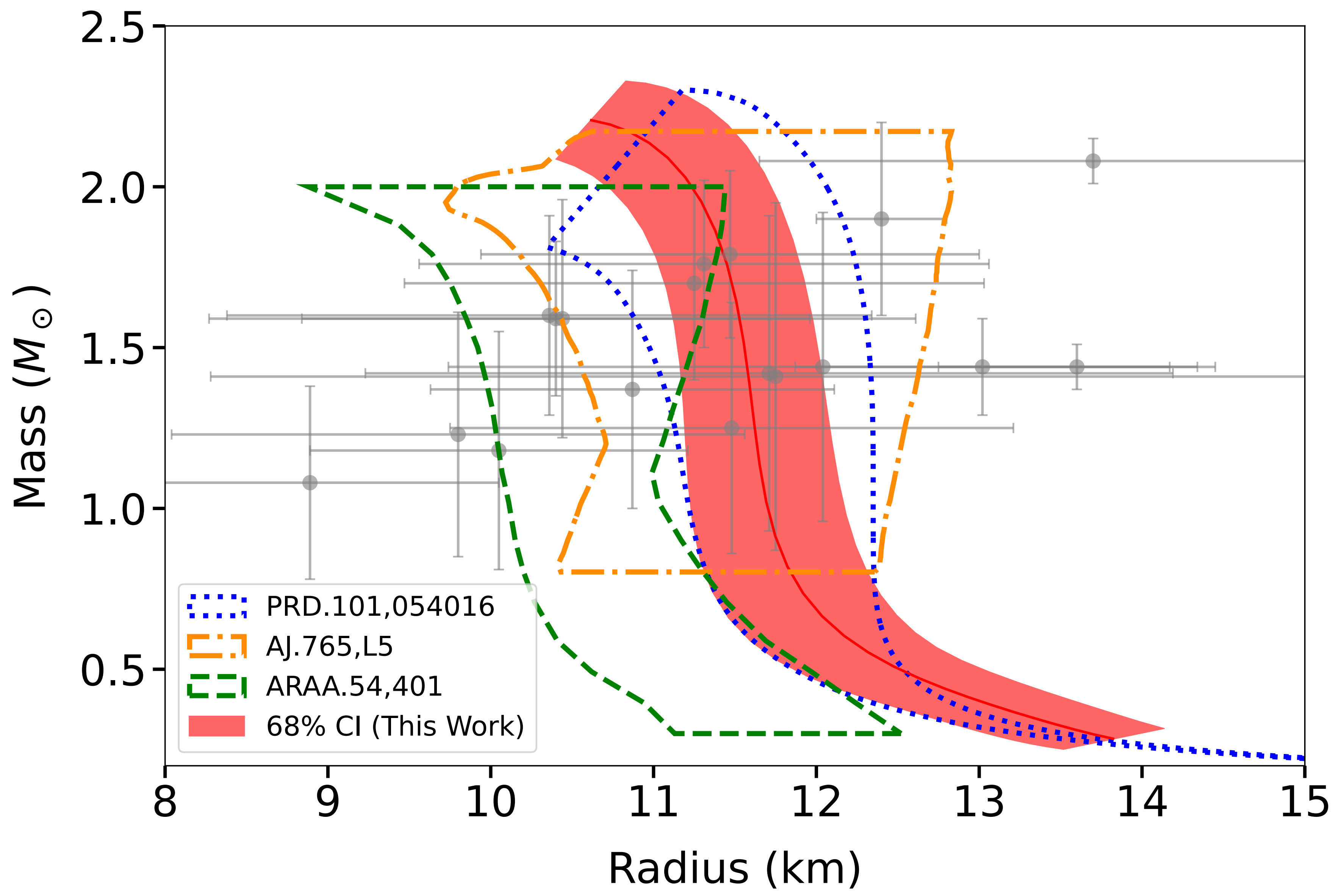
Physics-Driven Deep Learning

1. Building Neutron Star EoS

Phys. Rev. D 107, 083028

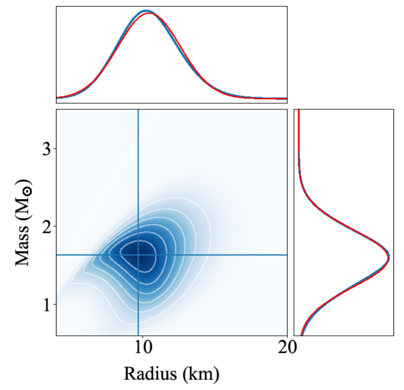
Blue dots: NN results, Fujimoto-Fukushima-Murase
Yellow and Green dashed lines: Bayesian Approaches





Our results: $R_{1.4} = 11.6 \pm 0.43$ **km (68% CI)**

18 (M_i, R_i) , sample size = **10k**
causality $(d\epsilon/dp < 1)$
Maximum mass $\geq 1.9\, M_\odot$



Observable	Mass($M_\odot$)	Radius(km)
M13	1.42±0.49	11.71±2.48
M28	1.08±0.30	8.89±1.16
M30	1.44±0.48	12.04±2.30
NGC 6304	1.41±0.54	11.75±3.47
NGC 6397	1.25±0.39	11.48±1.73
ωCen	1.23±0.38	9.80±1.76
4U 1608-52	1.60±0.31	10.36±1.98
4U 1724-207	1.79±0.26	11.47±1.53
4U 1820-30	1.76±0.26	11.31±1.75
EXO 1745-248	1.59±0.24	10.40±1.56
KS 1731-260	1.59±0.37	10.44±2.17
SAX J1748.9-2021	1.70±0.30	11.25±1.78
X5	1.18±0.37	10.05±1.16
X7	1.37±0.37	10.87±1.24
4U 1702-429	1.90±0.30	12.40±0.40
PSR J0437–4715	1.44±0.07	13.60±0.85
PSR J0030+0451	1.44±0.15	13.02±1.15
PSR J0740+6620	2.08±0.07	13.70±2.05

Physics-Driven Deep Learning

2. Reconstructing Spectral Function

Correlation Function

$$\bar{D}_{\eta\eta'} = \mathrm{Tr}(\mathrm{T}_\tau[J_\eta(\tau,\mathrm{x})J^\dagger_{\eta'}(0,0)]e^{-H/T})/Z$$

$$\equiv T\sum_n \int \frac{d^3k}{(2\pi)^3} D_{\eta\eta'}(i\omega_n,\mathrm{k})e^{-i(\omega_n\tau-\mathrm{k}\cdot\mathrm{x})}$$

M. Asakawa, Y. Nakahara, and T. Hatsuda, Prog. Part. Nucl. Phys. 46, 459 (2001)

$\mathbf{k}=0, \eta=\eta'$

$\mathbf{k}=0, \eta=\eta', T\to 0$

Spectrum representation

$$G(\tau,T) = \int_0^\infty K(\omega,\tau,T)\rho(\omega,T)d\omega$$

$$K(\omega,\tau,T) = \frac{\cosh\omega(\tau-\frac{1}{2T})}{\sinh\frac{\omega}{2T}}$$

Thermal Kernel

$T\to 0$

Energy eigenspace

$$C(\tau) = \sum_{n=0} e^{-E_n\tau}\,|<\Omega_0\,|\,J(0)\,|\,\Omega_n>|^2$$

$$\equiv \sum_{n=0} |c_n|^2 e^{-E_n\tau}$$

Discretization

$$G(\tau) = \int_0^\infty e^{-\tau\omega}\rho(\omega)d\omega$$

Laplace Kernel

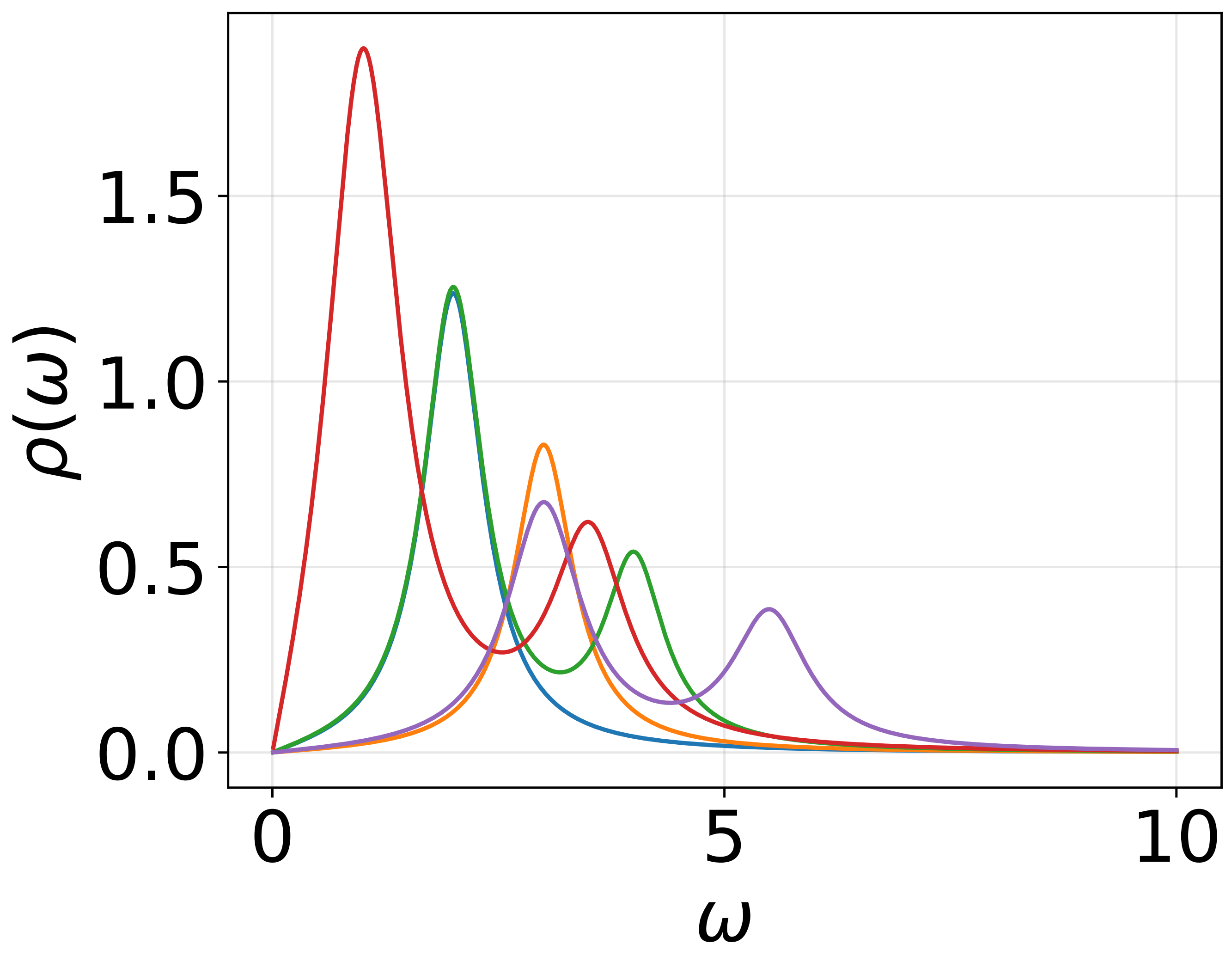
Spectral Function $\rho(\omega), \quad E_n$

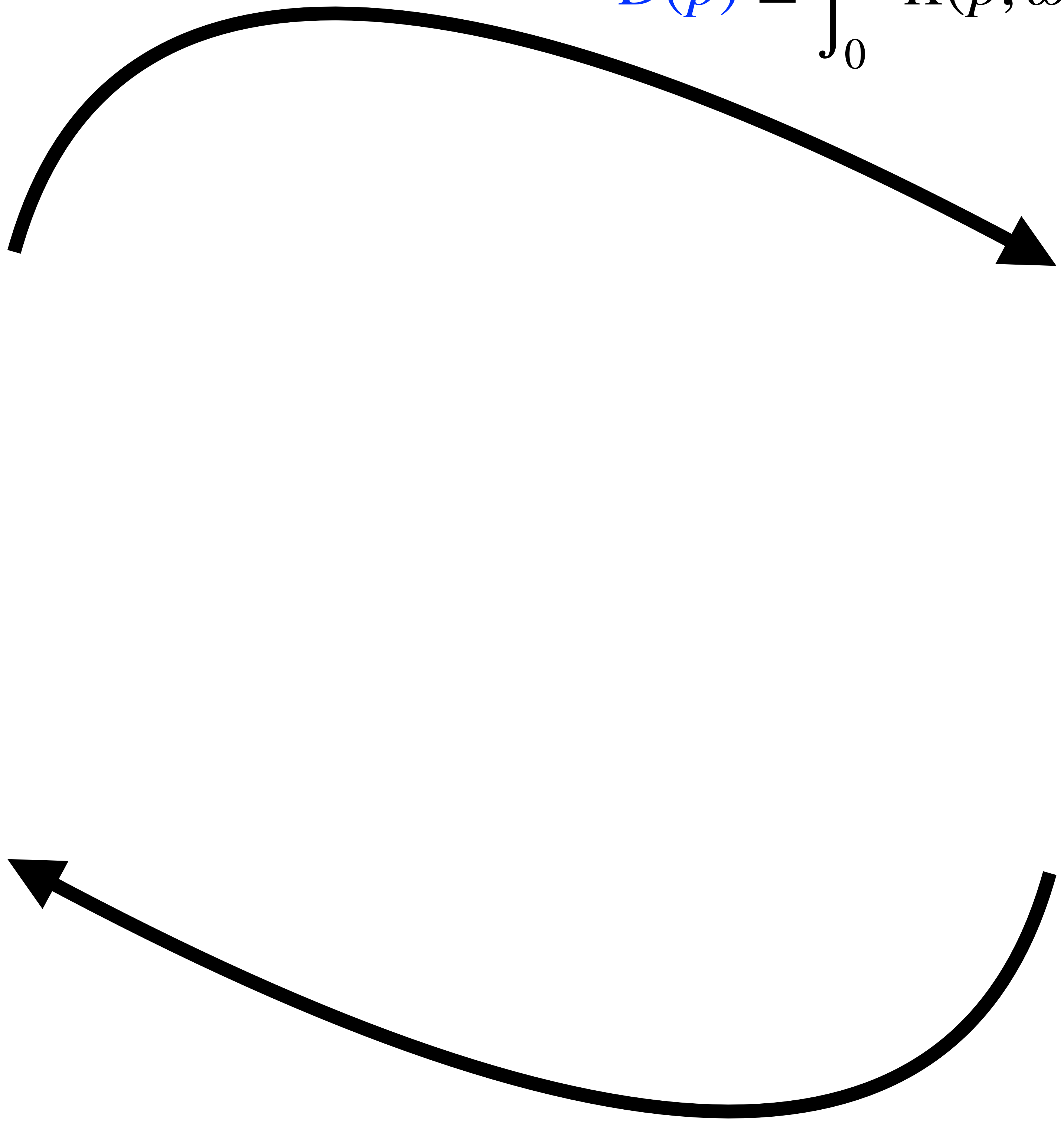
Physics-Driven Deep Learning

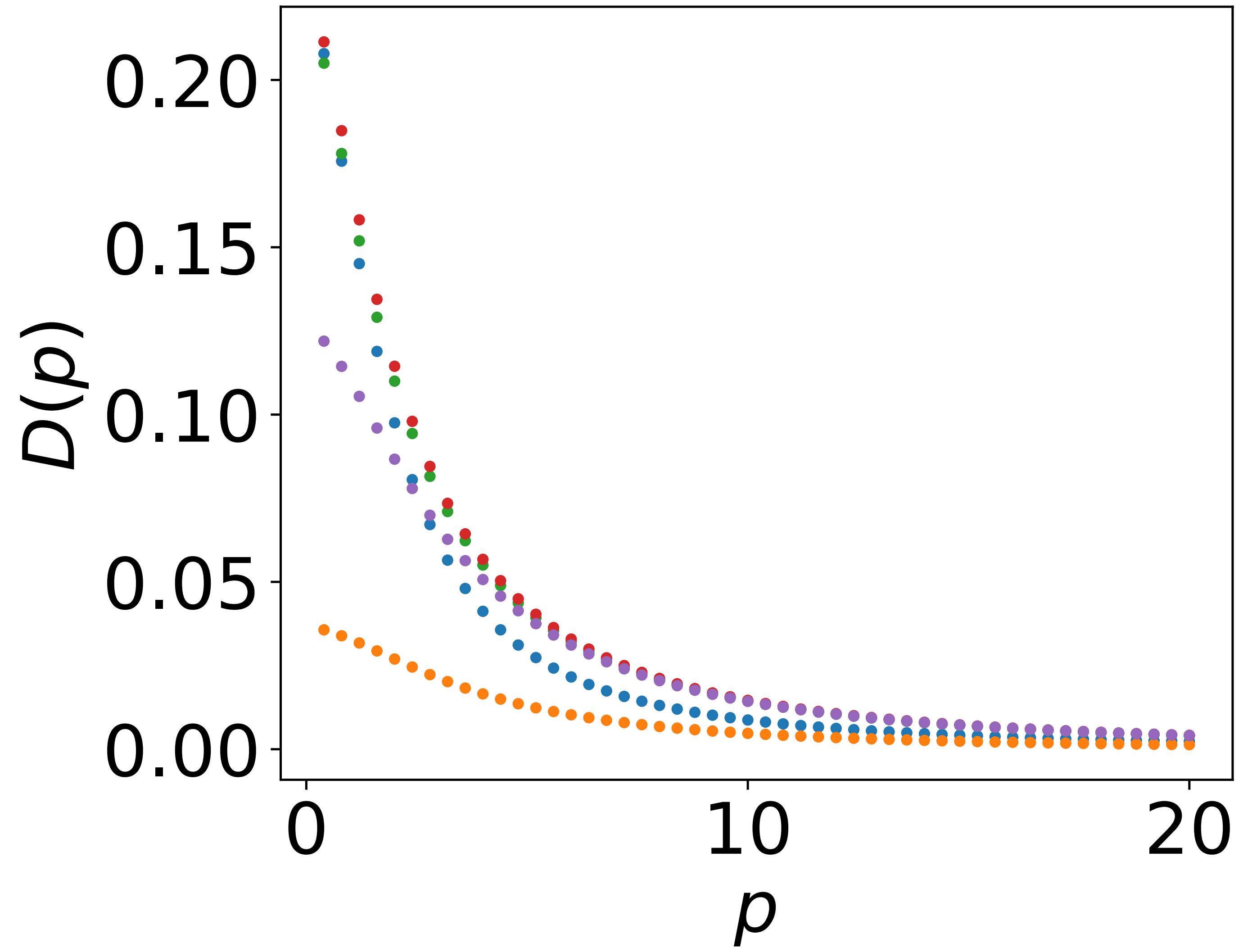
2. Reconstructing Spectral Function

***Kallen—Lehmann(KL)* representation**

$$D(p) \equiv \int_0^\infty K(p,\omega)\rho(\omega)\mathrm{d}\omega \quad K(p,\omega) = \frac{\omega}{\pi(\omega^2+p^2)}$$





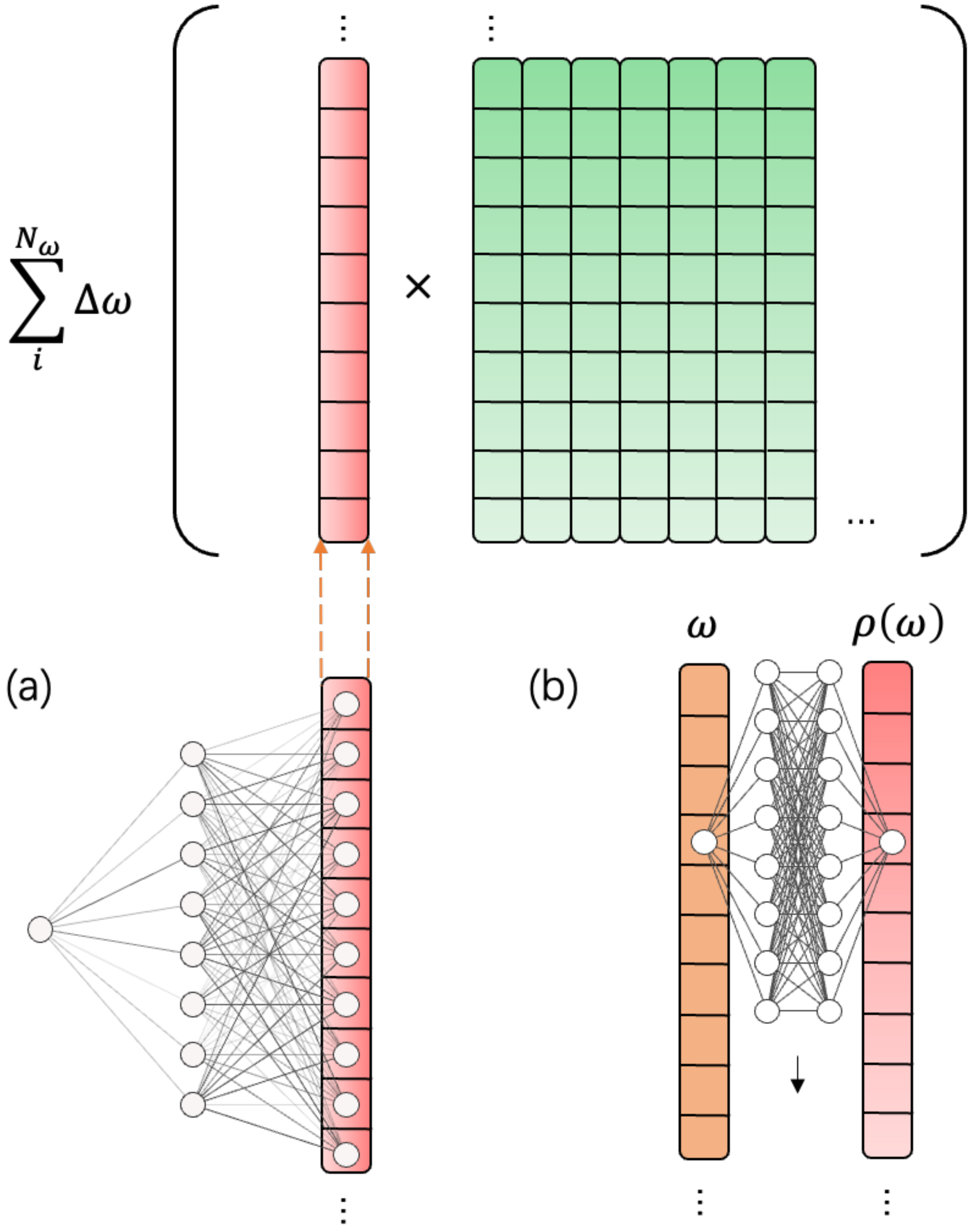


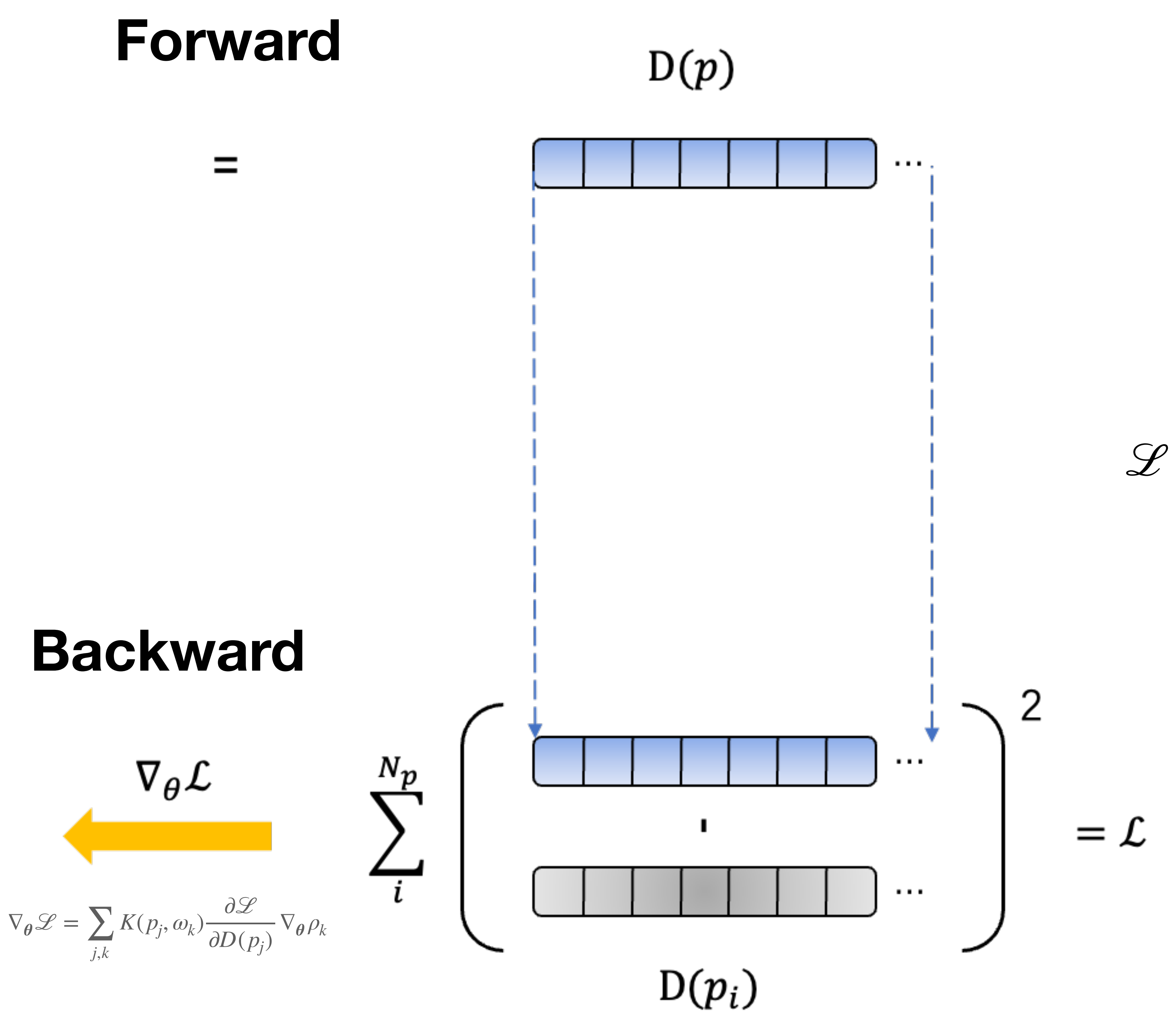
Physics-Driven Deep Learning

2. Reconstructing Spectral Function

Phys. Rev. D 106, L051502

$$\int_0^\infty K(p,\omega)\rho_\theta(\omega)\mathrm{d}\omega = D(p)$$





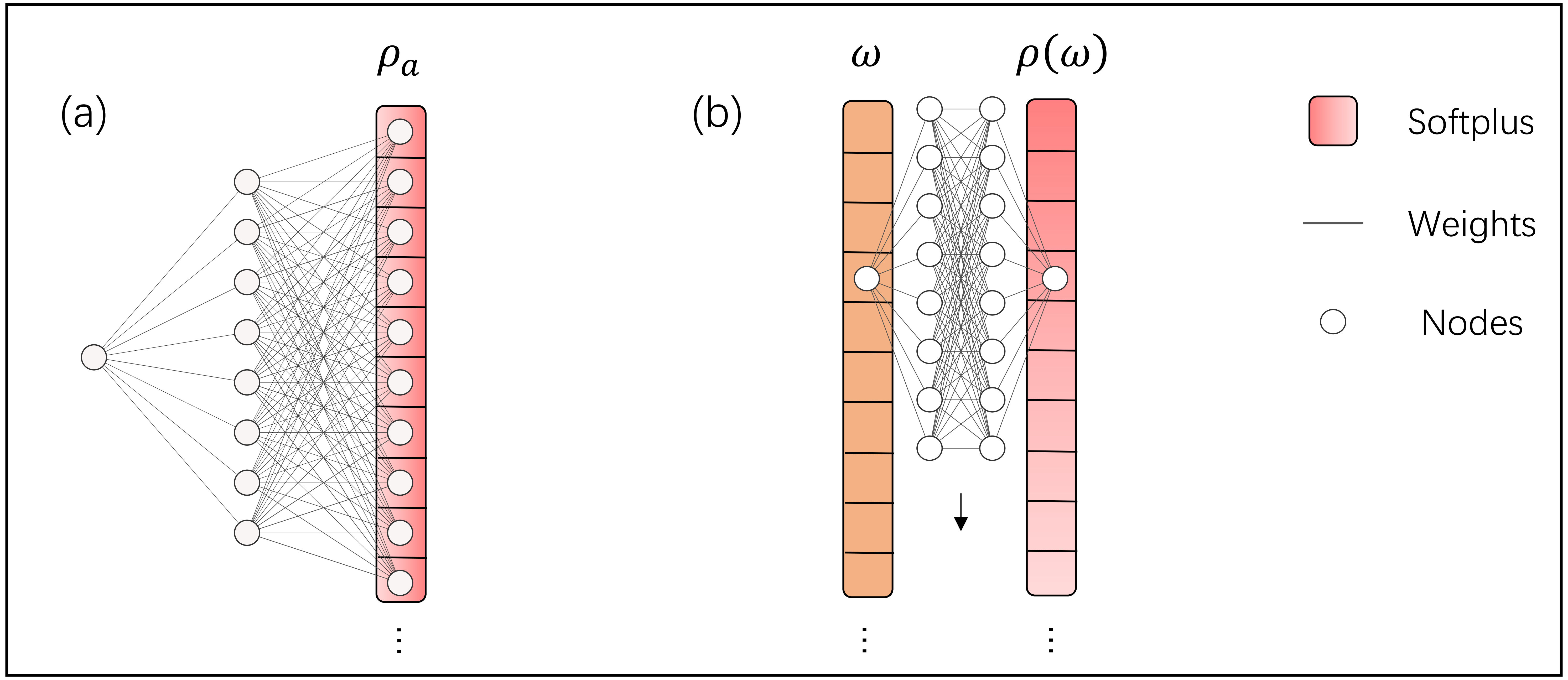
χ^2 Loss function

$$\mathscr{L} = \sum_i^{N_p} w_i(\mathrm{D}_i - D(p_i))^2$$

Physics-Driven Deep Learning

2. Reconstructing Spectral Function

Phys. Rev. D 106, L051502



NN : $(\rho_1, \rho_2, \cdots, \rho_{N_\omega})$

Differentiable variables : Network weights $\{\theta\}$

Adam, L2 ($\lambda = 10^{-3} \rightarrow 10^{-8}$), Smoothness ($\lambda_s = 10^{-4} \rightarrow 0$)

width = 64 and depth = 3 with bias

NN-P2P : $\rho(\omega)$

Differentiable variables : Network weights $\{\theta\}$

Adam, L2 ($\lambda = 10^{-6} \rightarrow 0$)

width = 64 and depth = 3 with bias

Regularization

L2 : $\lambda \mid \theta \mid_2^2$

Smoothness: $\lambda_s \sum_i^{N_\omega} (\rho_i - \rho_{i-1})^2$

Gradient-based Optimization

Adam : $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$

Physical Prior

Positive-defined condition(for hadrons):
Softplus $log(1 + e^x)$

Physics-Driven Deep Learning

2. Reconstructing Spectral Function

Phys. Rev. D 106, L051502

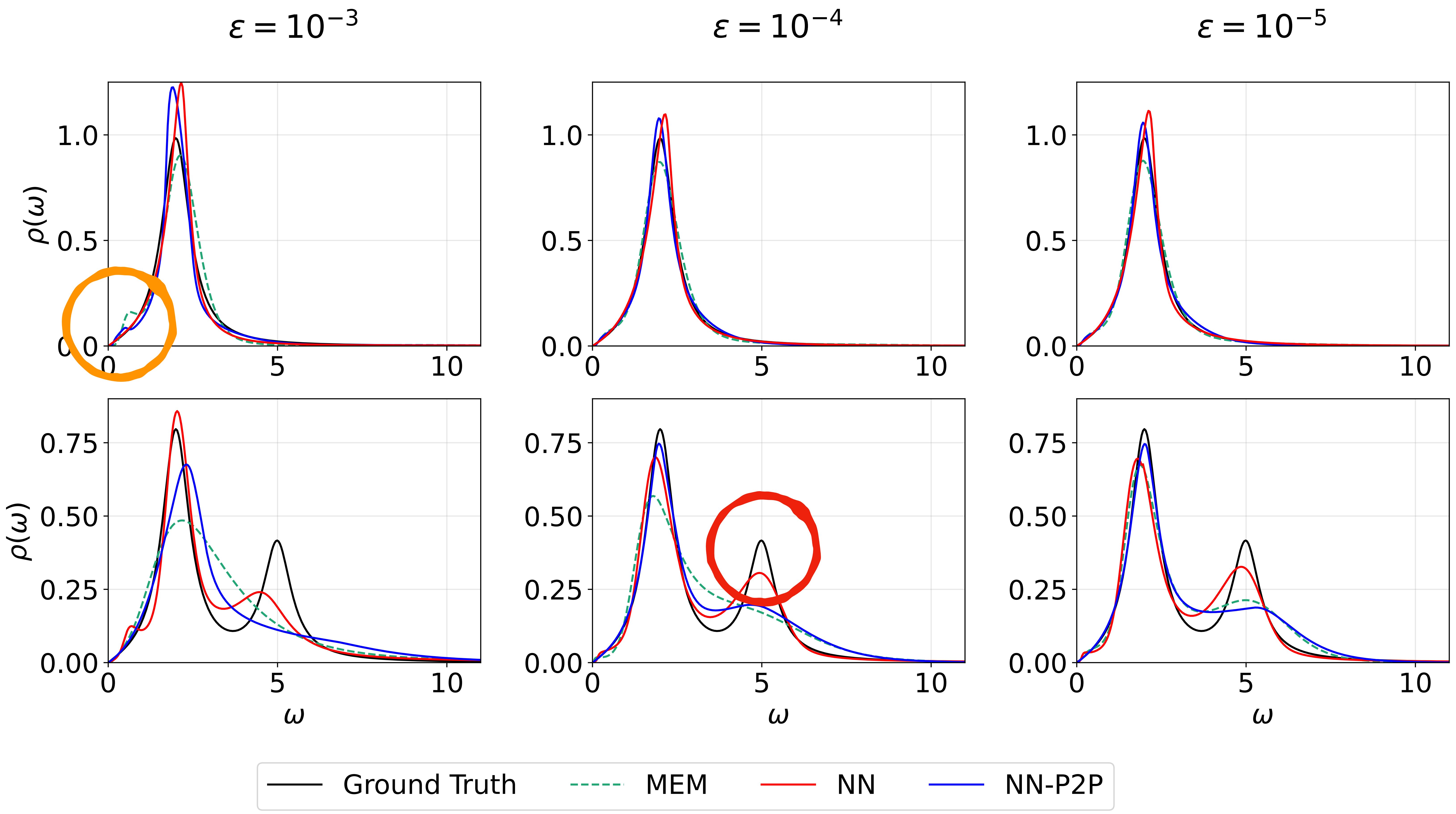
Mock Test I.

$$\rho^{(\mathrm{BW})}(\omega) = \frac{4A\Gamma\omega}{\left(M^2+\Gamma^2-\omega^2\right)^2+4\Gamma^2\omega^2}$$

$A = 1.0, \Gamma = 0.5, M = 2.0$

$A_1 = 0.8, A_2 = 1.0, \Gamma_1 = \Gamma_2 = 0.5$
$M_1 = 2.0, M_2 = 5.0$

noise level ϵ
$N_p = 25$ points



Physics-Driven Deep Learning

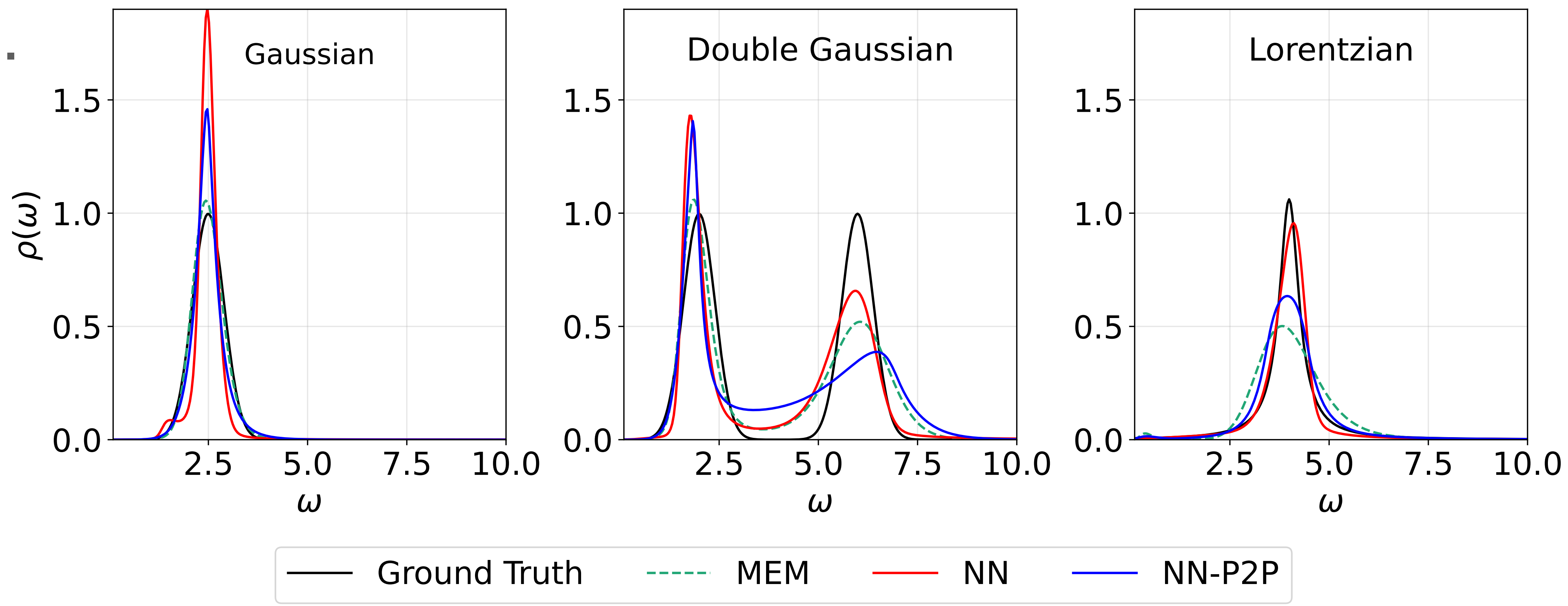
2. Reconstructing Spectral Function

Phys. Rev. D 106, L051502

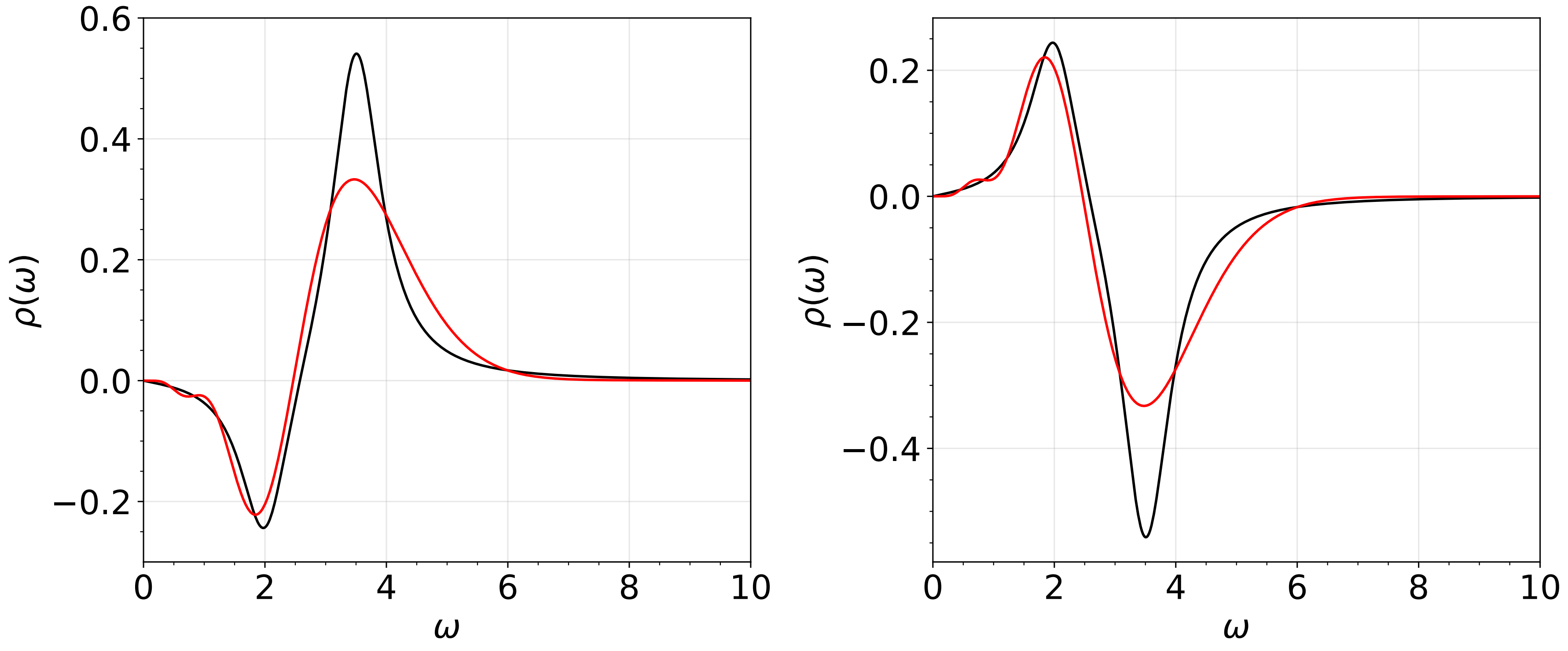
Mock Test II.

noise level $\epsilon = 10^{-4}$ with $N_p = 25$ points

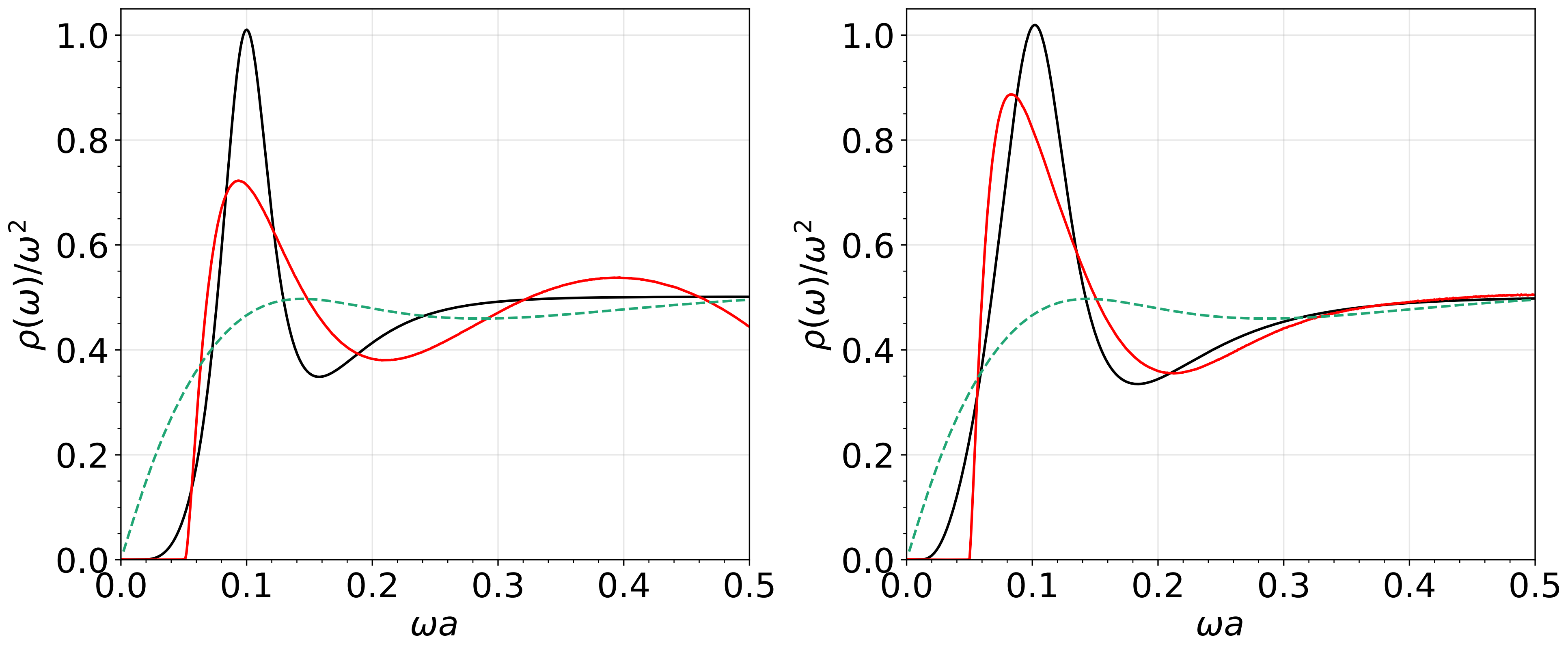
1.



2.



3.



1. Single-peak functions

2. Non-positive-definited SPs

3. Lattice QCD mock data

Thermal (details see arXiv:2110.13521)

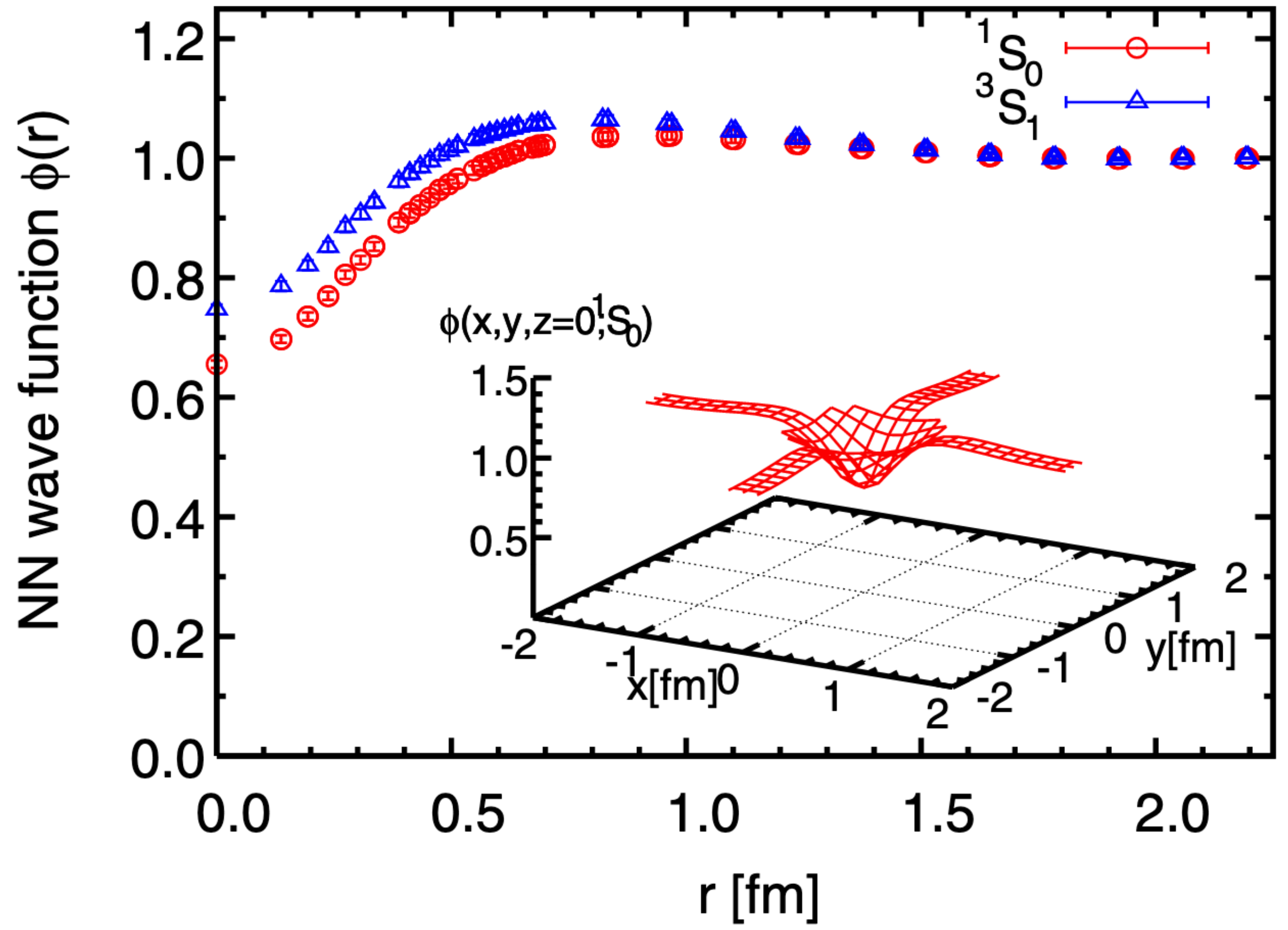
$$G(\tau, T) = \int_0^\infty \frac{d\omega}{2\pi} K(\omega, \tau, T)\rho(\omega, T)$$

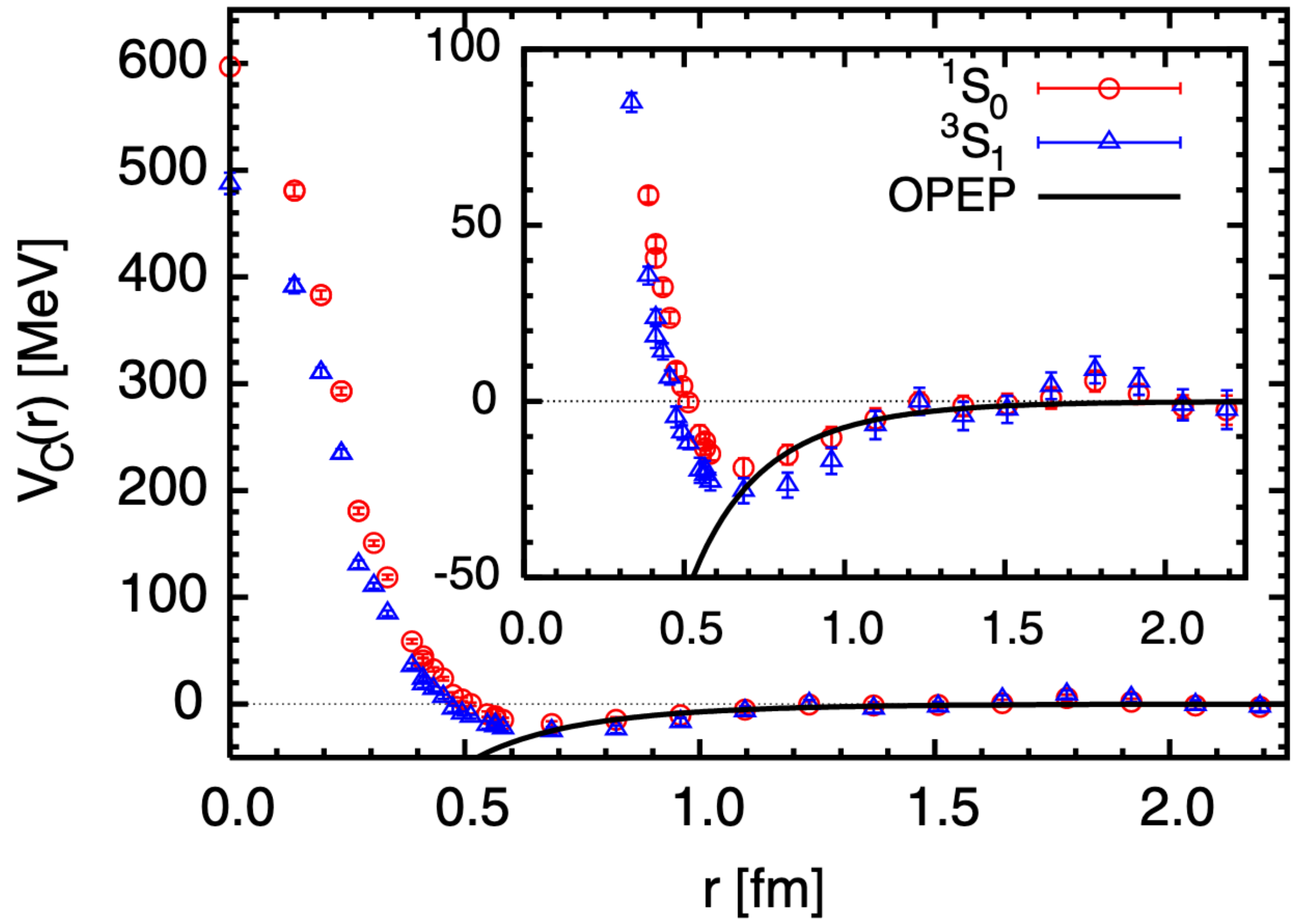
$$K(\omega, \tau, T) = \frac{\cosh \omega(\tau - \frac{1}{2T})}{\sinh \frac{\omega}{2T}}$$

Physics-Driven Deep Learning

3. Extracting Nuclear Force

N. Ishii, S. Aoki, and T. Hatsuda, Phys. Rev. Lett. **99**, 022001 (2007)





Local Approx.
Gradient Expansion

Nambu-Bethe-Salpeter (NBS) wave function → **Nulcear Force**

HAL QCD method

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

(at asymptotic region)

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

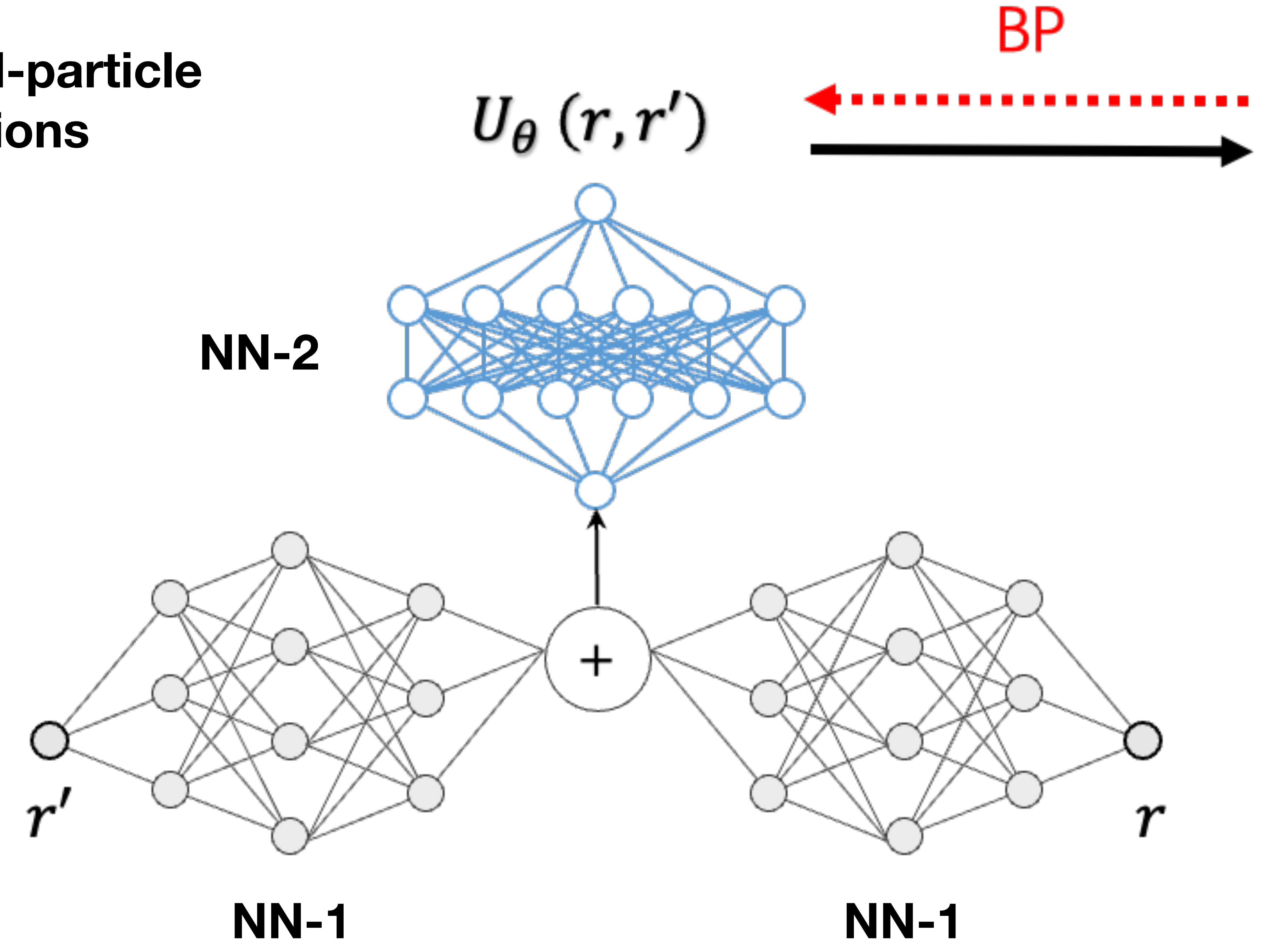
(Schrodinger eq.)

Physics-Driven Deep Learning

3. Extracting Nuclear Force

in preparation (with HAL QCD)

Two identical-particle interactions



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)

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

Physics-Driven Deep Learning

in preparation (with HAL QCD)

3. Extracting Nuclear Force

Separable Potential

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

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

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

Nambu-Bethe-Salpeter (NBS) wave function

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

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

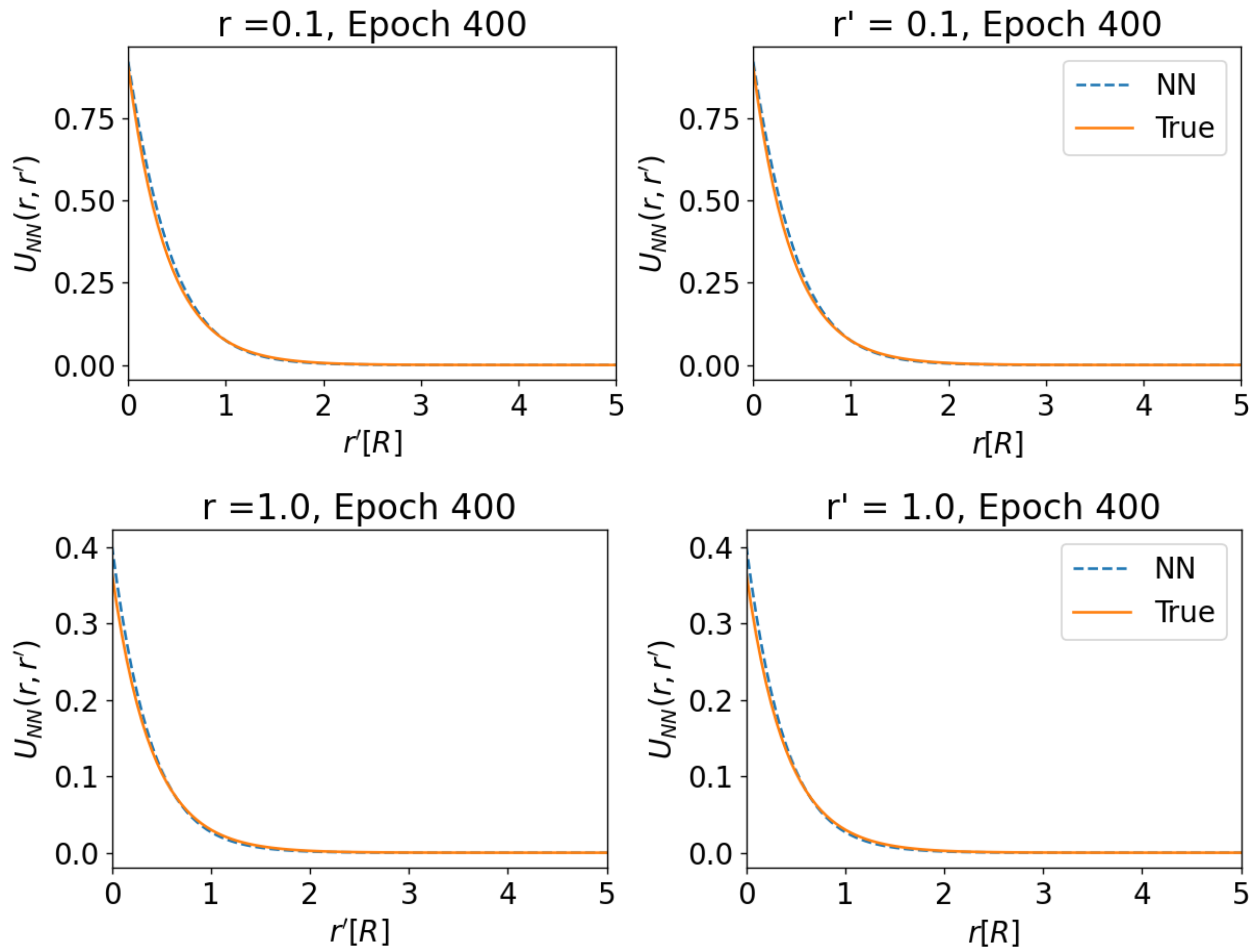
$$E_k = \frac{k^2}{2m}, \quad H_0 = -\frac{\nabla^2}{2m}, \quad m = \frac{m_N}{2}$$

No Approx.

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

Neural Network Hadron Force

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



Physics-Driven Deep Learning

3. Extracting Nuclear Force

in preparation (with HAL QCD)

$\Omega_{ccc}\Omega_{ccc}$ **Interaction**

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

Nambu-Bethe-Salpeter (NBS) wave function

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

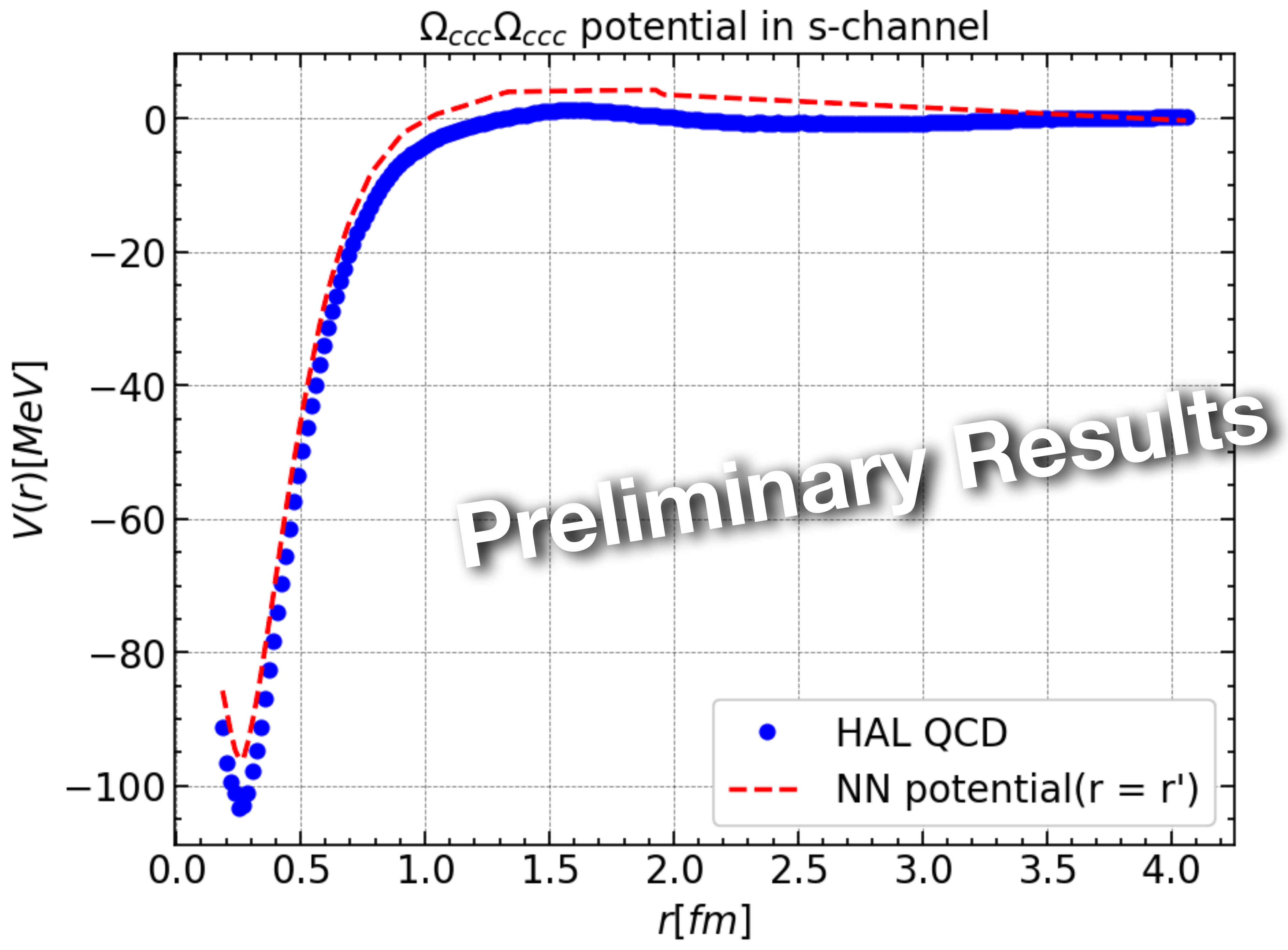
$$m_N = 2.073, a^{-1} = 2333.0\textbf{MeV}$$

$$t = 26$$

Phys. Rev. Lett. 127, 072003 (2021)

No Approx.

Neural Network Hadron Force



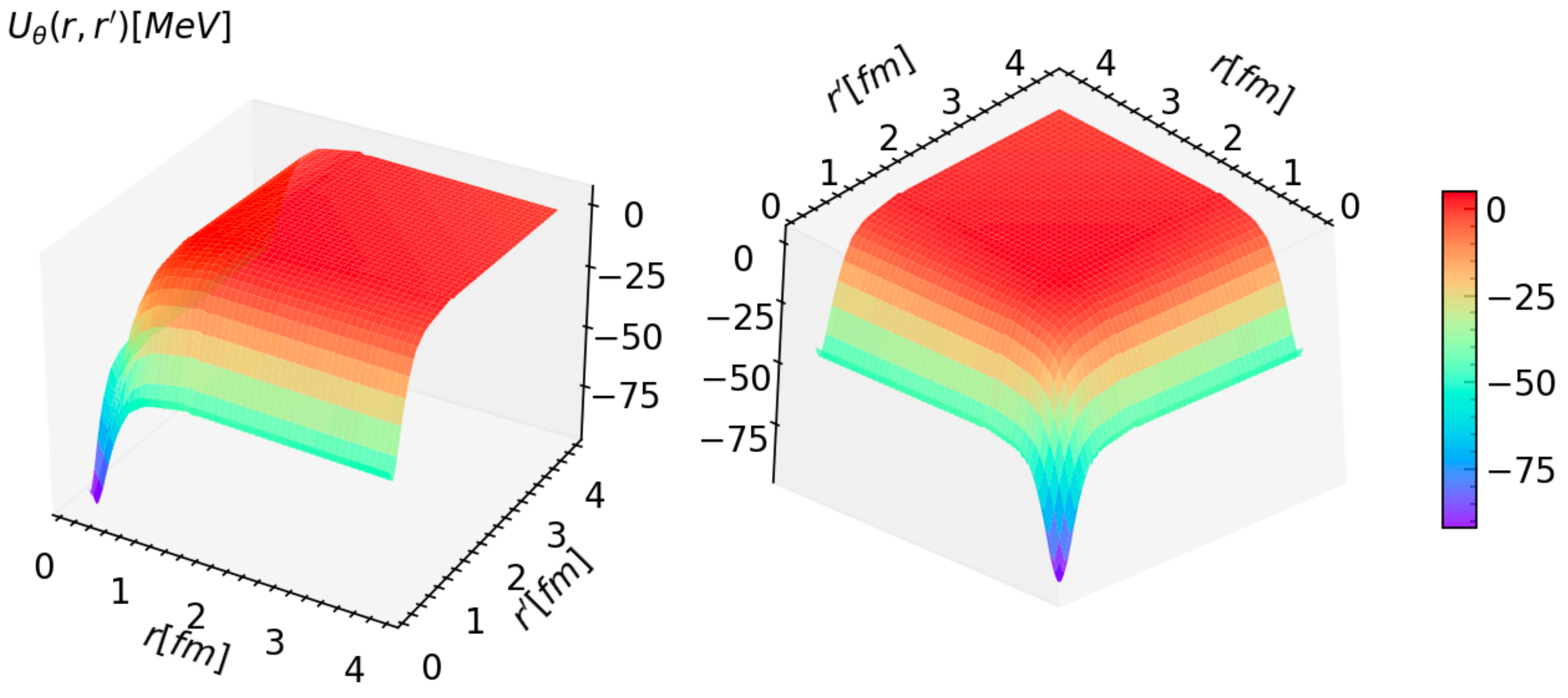
$$\mathcal{L}=\sum_t\left\{\frac{1}{4m_N}R2(t,r)-R1(t,r)+\frac{1}{m_N}Rr(t,r)-\int 4\pi r'^2 dr'\, U_\theta(r,r')R(t,r')\right\}$$

Physics-Driven Deep Learning

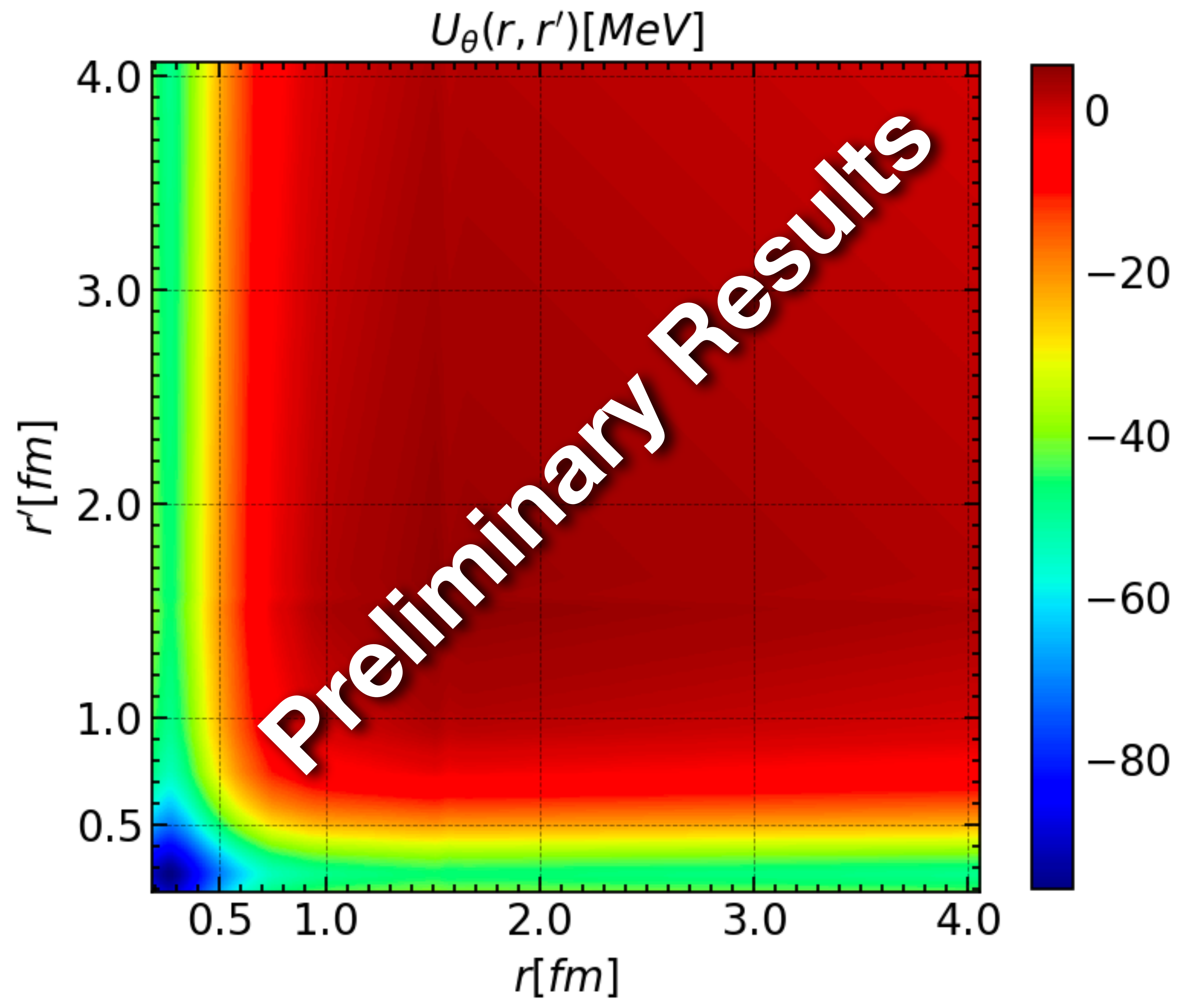
3. Extracting Nuclear Force

in preparation (with HAL QCD)

$\Omega_{ccc}\Omega_{ccc}$ **Interaction**

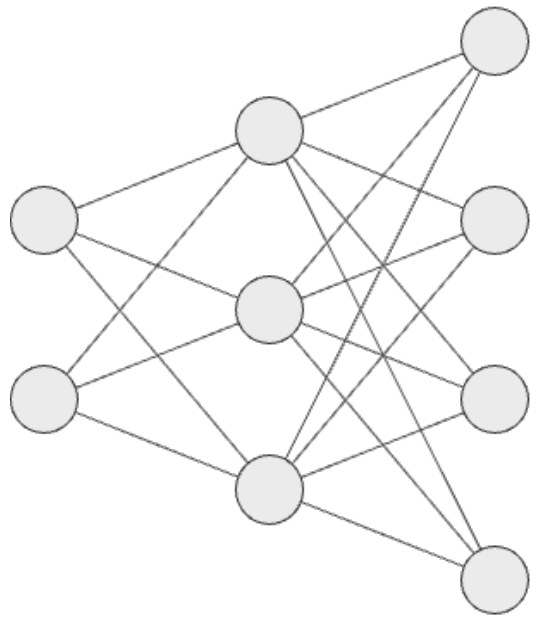


Neural Network Hadron Force



Summary I

- **Inverse Problems**
  - Data-driven learning
  - Physics-driven learning
  - Physics-driven **deep** learning
    - Neural network representations
    - Gradient-based optimization
- **Future works**
  - Nuclear Matter EoS
  - Spectroscopy [github1, github2]
  - NN-Nulcear Force



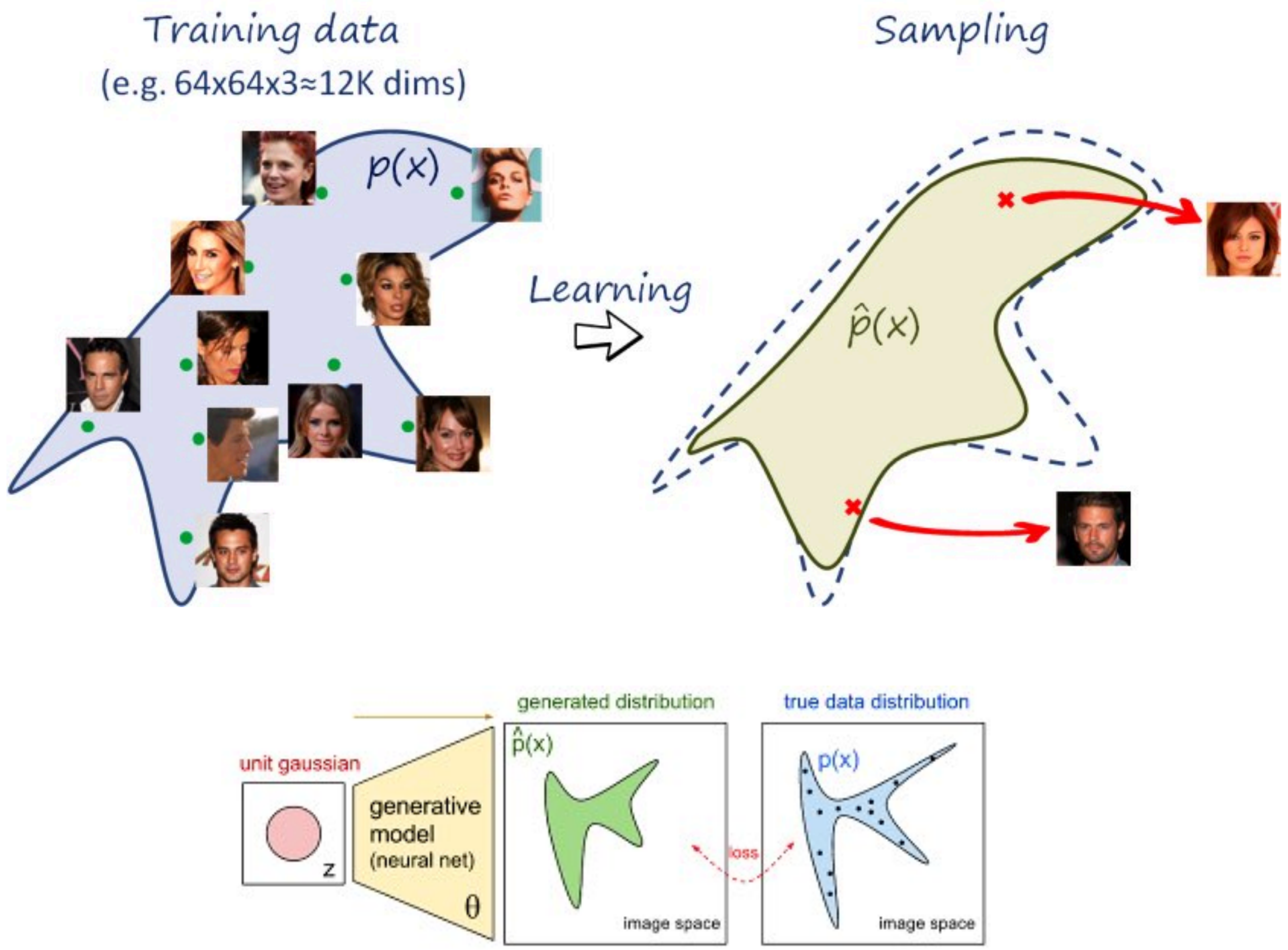
$$f_\theta \xrightarrow{\text{Phys}} \tilde{O}$$

$$f_\theta \xleftarrow{\text{BP}} \mathcal{L} = (O - \tilde{O})^2$$

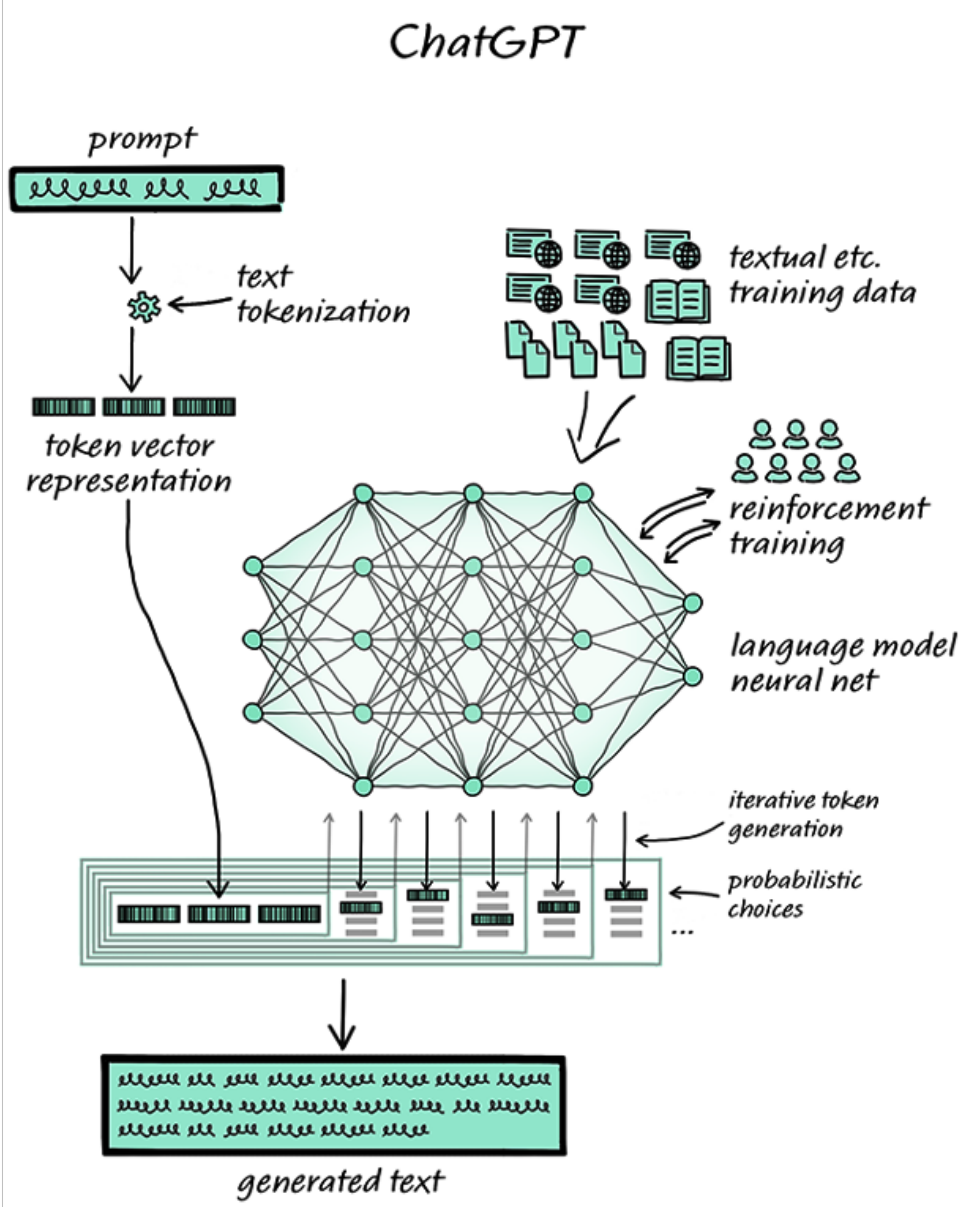
$$\hat{\theta} = \mathbf{arg\ max}_{\theta}\{p(X \mid \theta)\}$$

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

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

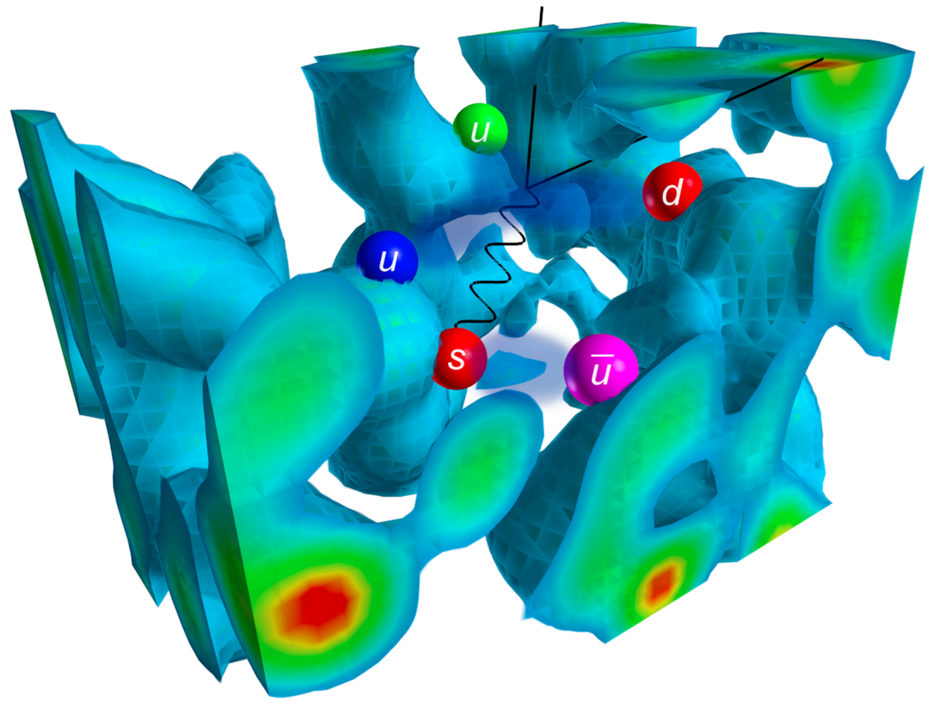
$$\langle O \rangle \approx \frac{1}{N}\sum_i O_i$$

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

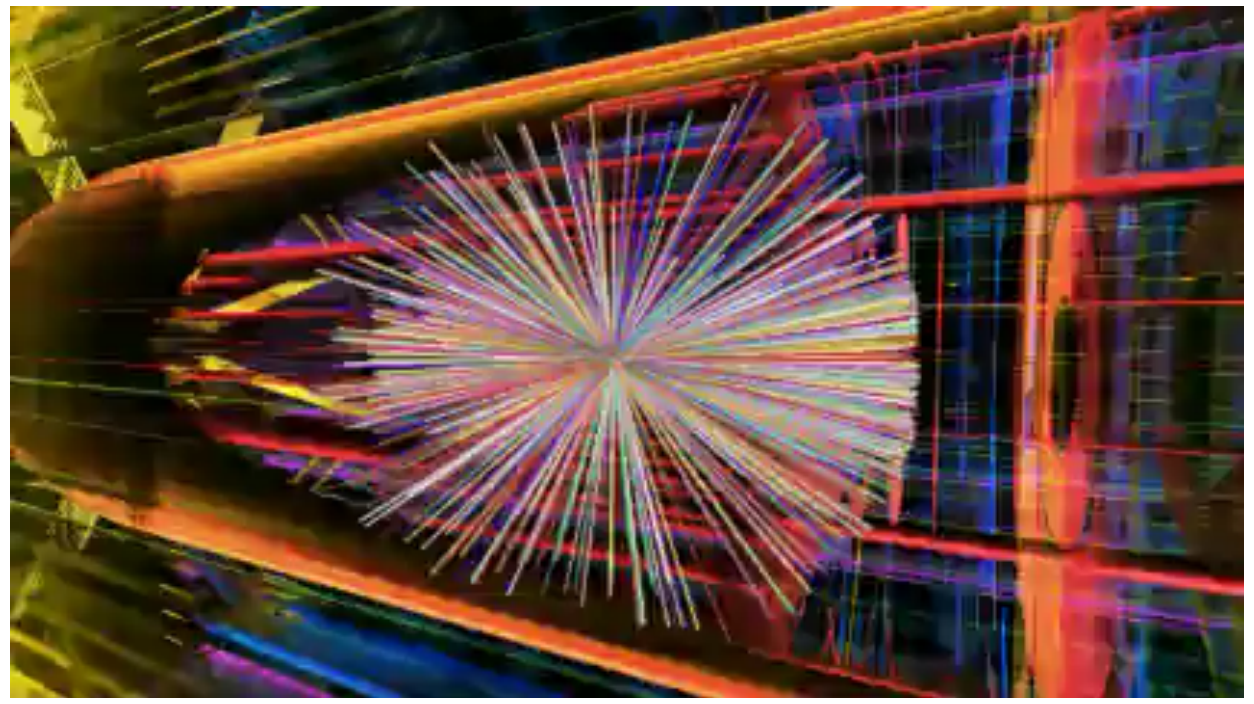
Global Sampling

Fast and Independent Sampler

Prog.Part.Nucl.Phys. 104084(2023) **(Invited Review)**



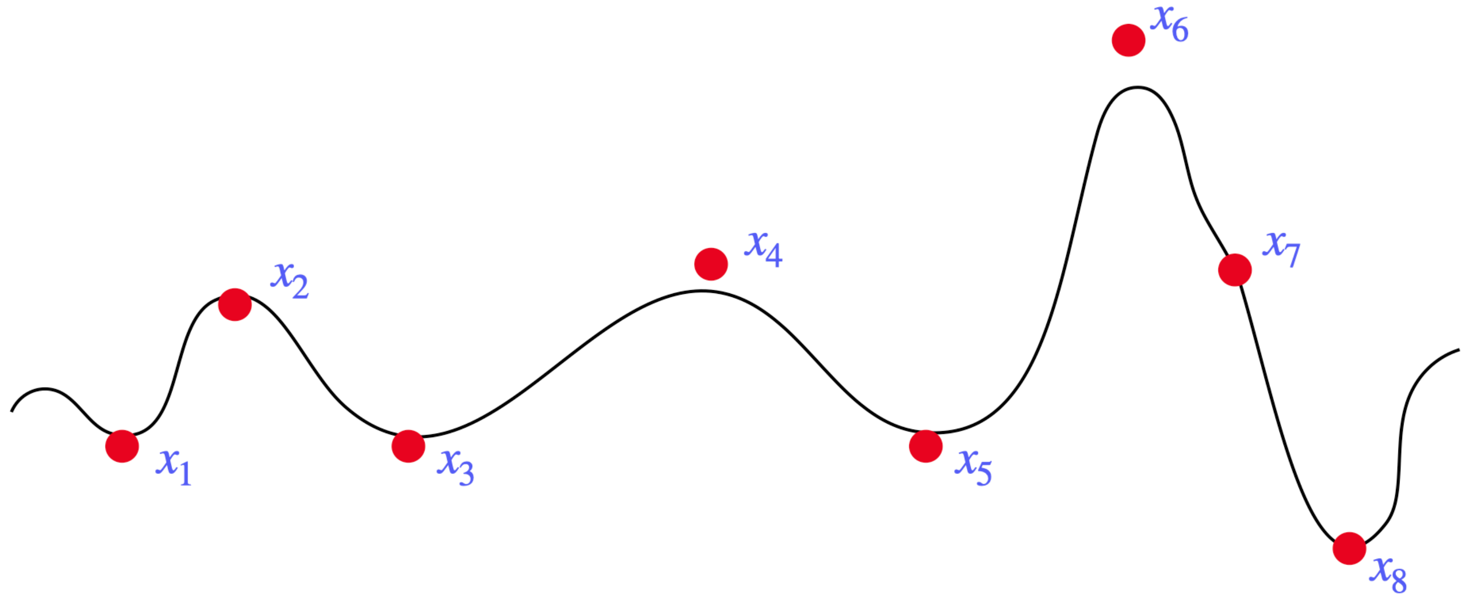
Lattice QCD © Derek Leinweber/CSSM/University of Adelaide



Heavy-Ion Collisions © 2010 CERN

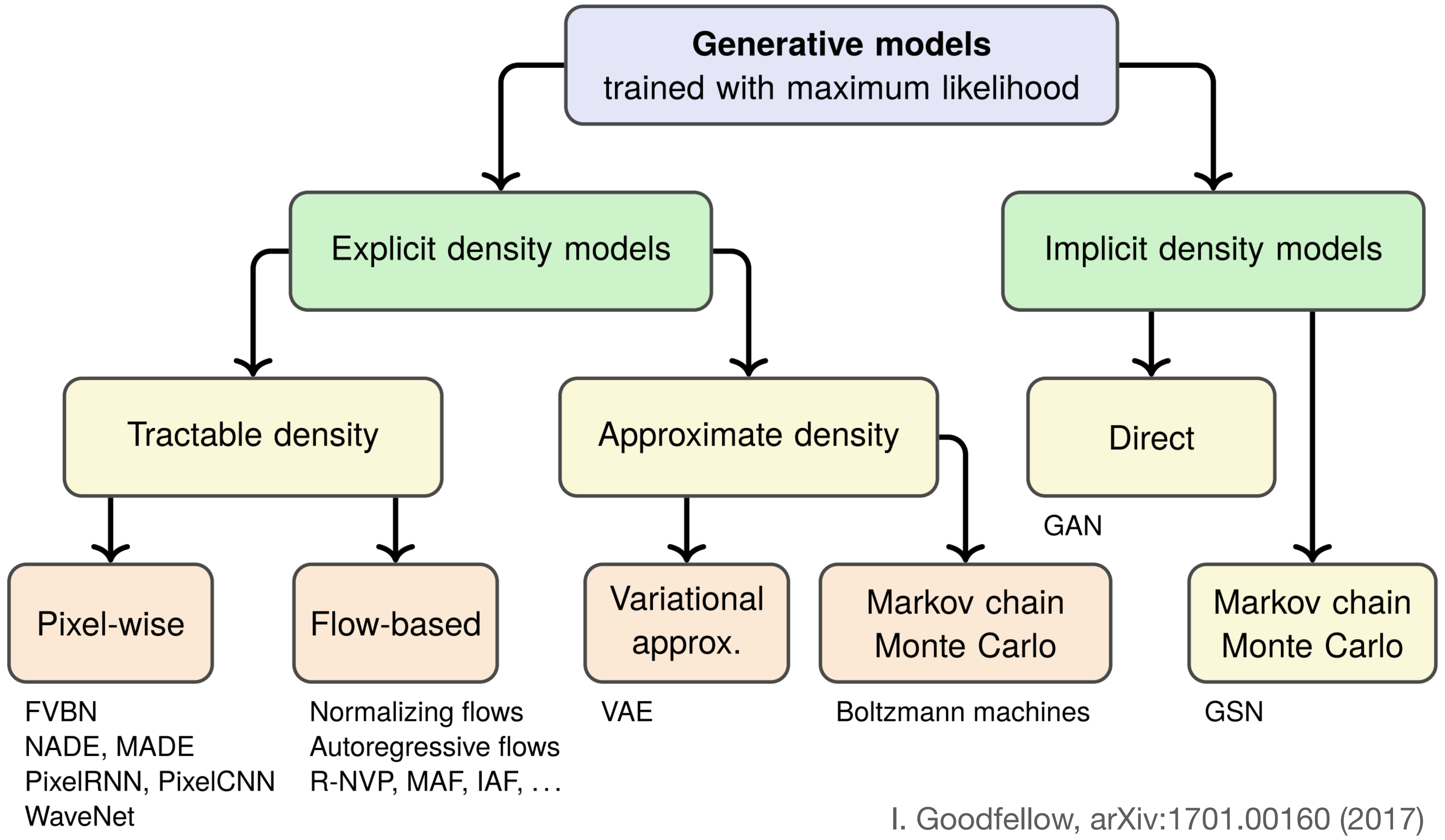
Probabilistic Models

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



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

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

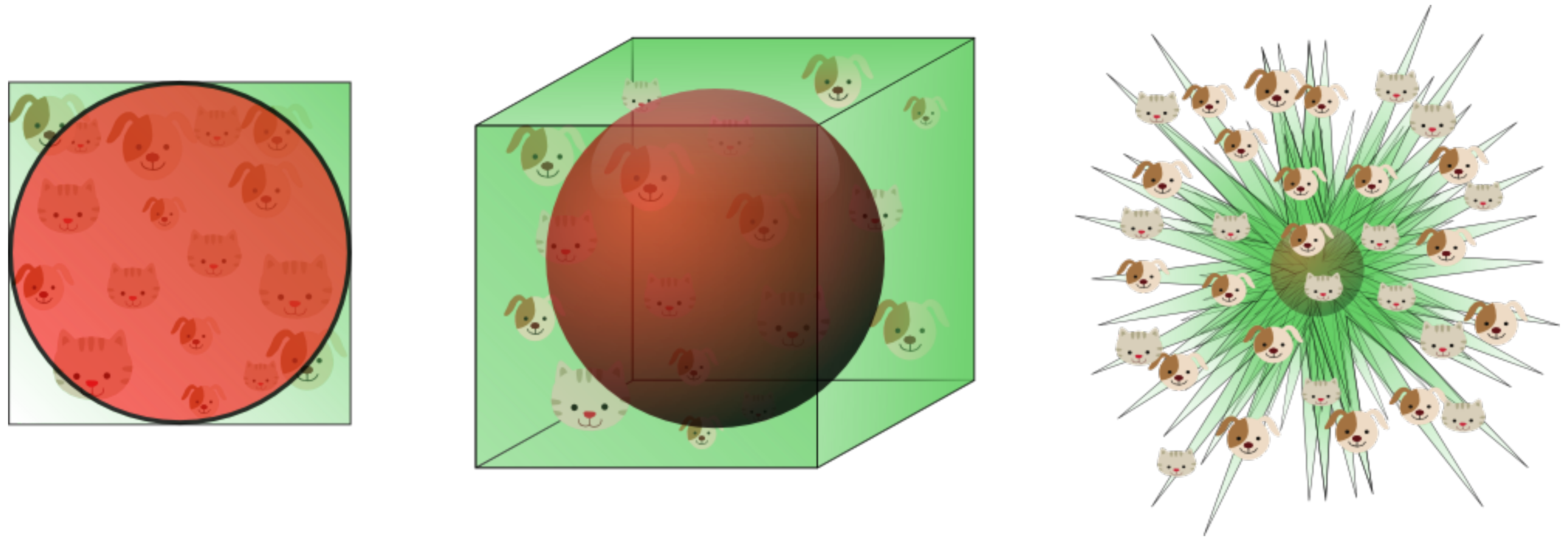


I. Goodfellow, arXiv:1701.00160 (2017)

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

Probabilistic Models

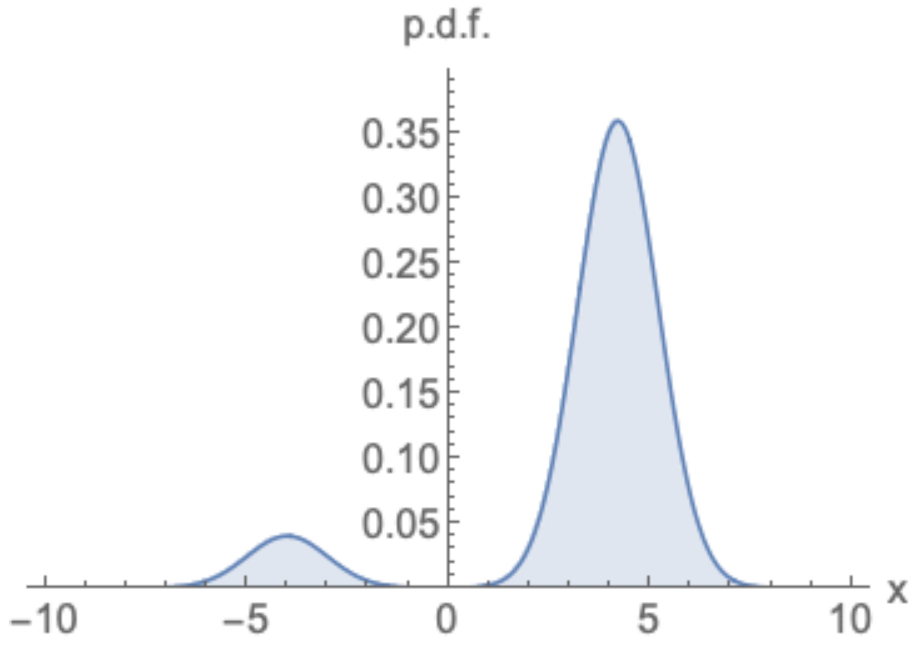
$$\mathbf{x} \sim p_{data}(\mathbf{x})$$



Curse of Dimensionality

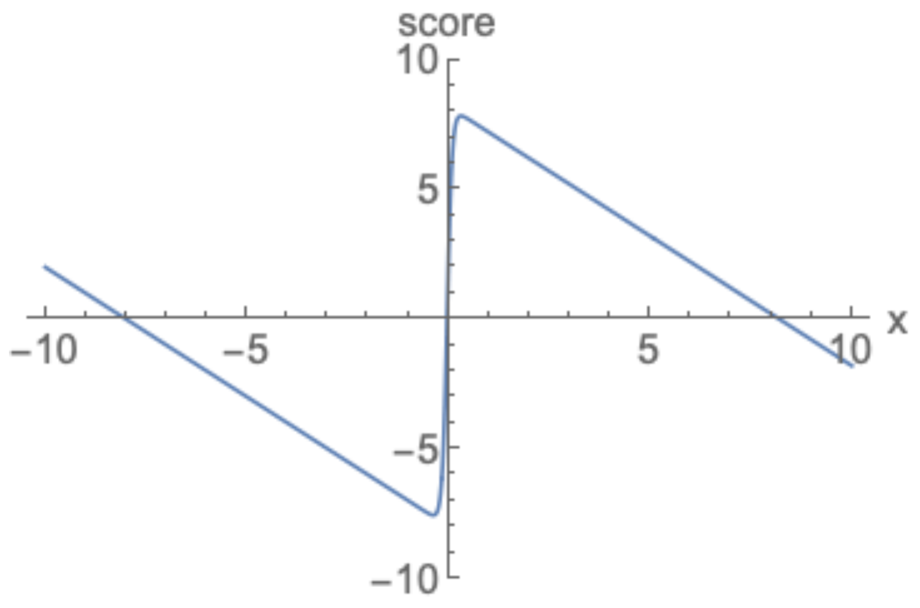
$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$$

from a **statistical phsyics** perspective



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

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



Approaching **Score Function**

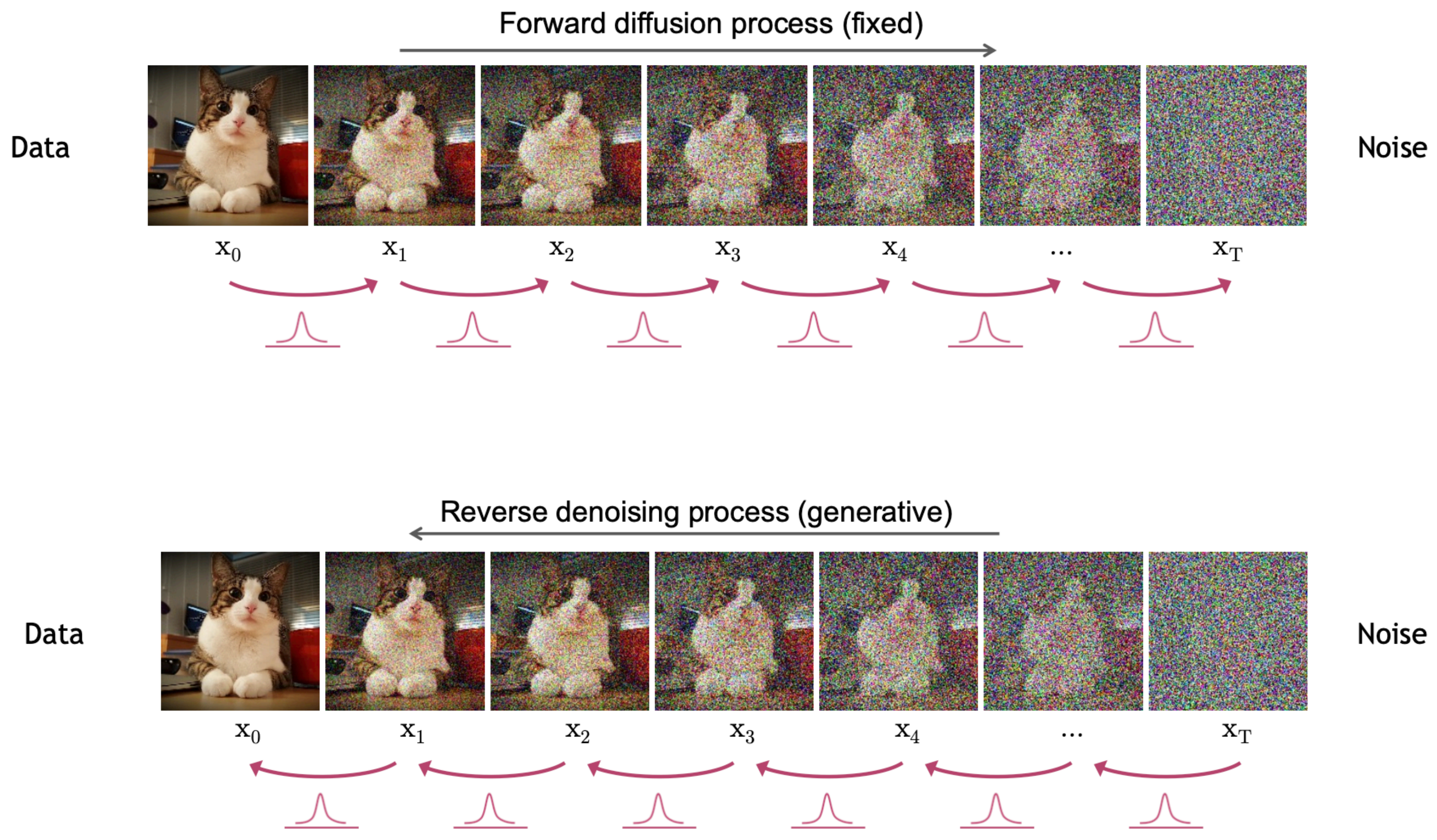
$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_\mathbf{x} \log p_\theta(\mathbf{x})$$

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

Space Opera, Jason Allen via Midjourney

Diffusion Models

- Forward diffusion process **gradually adds noise** to input
- Reverse denoising process learns to **generate data by denoising**
- Train **Probabilistic Models** to learn how to **convert a simple distribution to a target distribution**



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

Diffusion Models

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

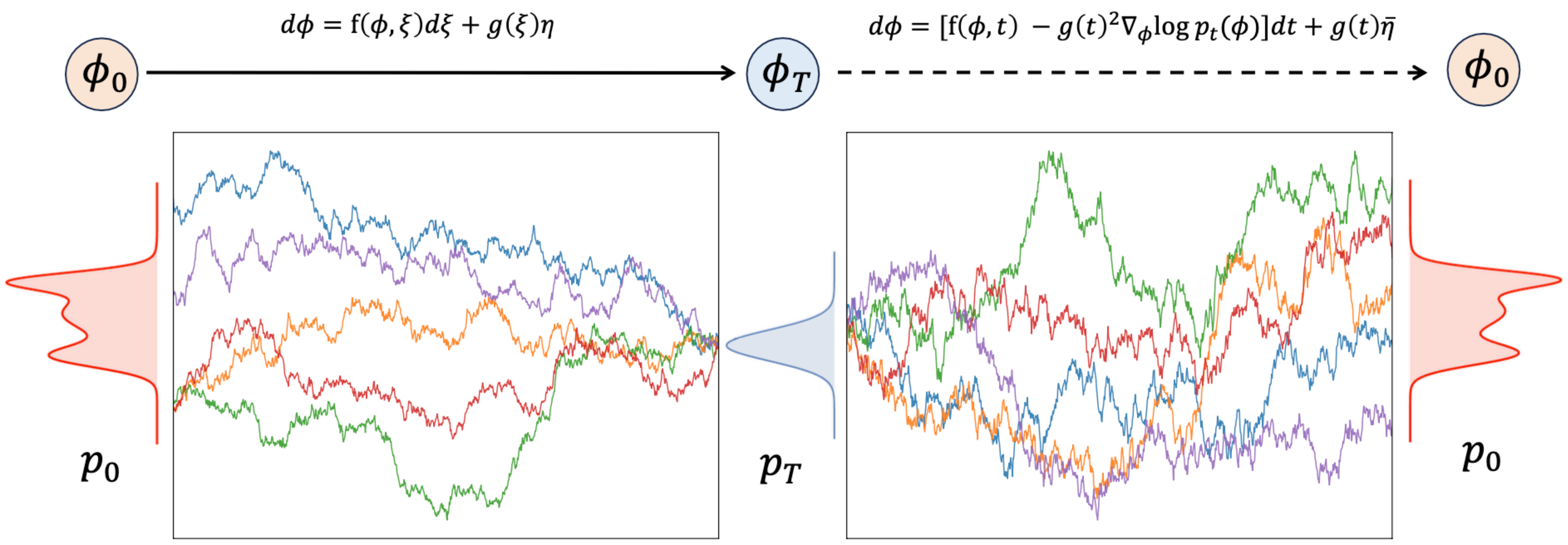
Anderson, in Stochastic Processes and their Applications, 1982

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

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

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

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



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**

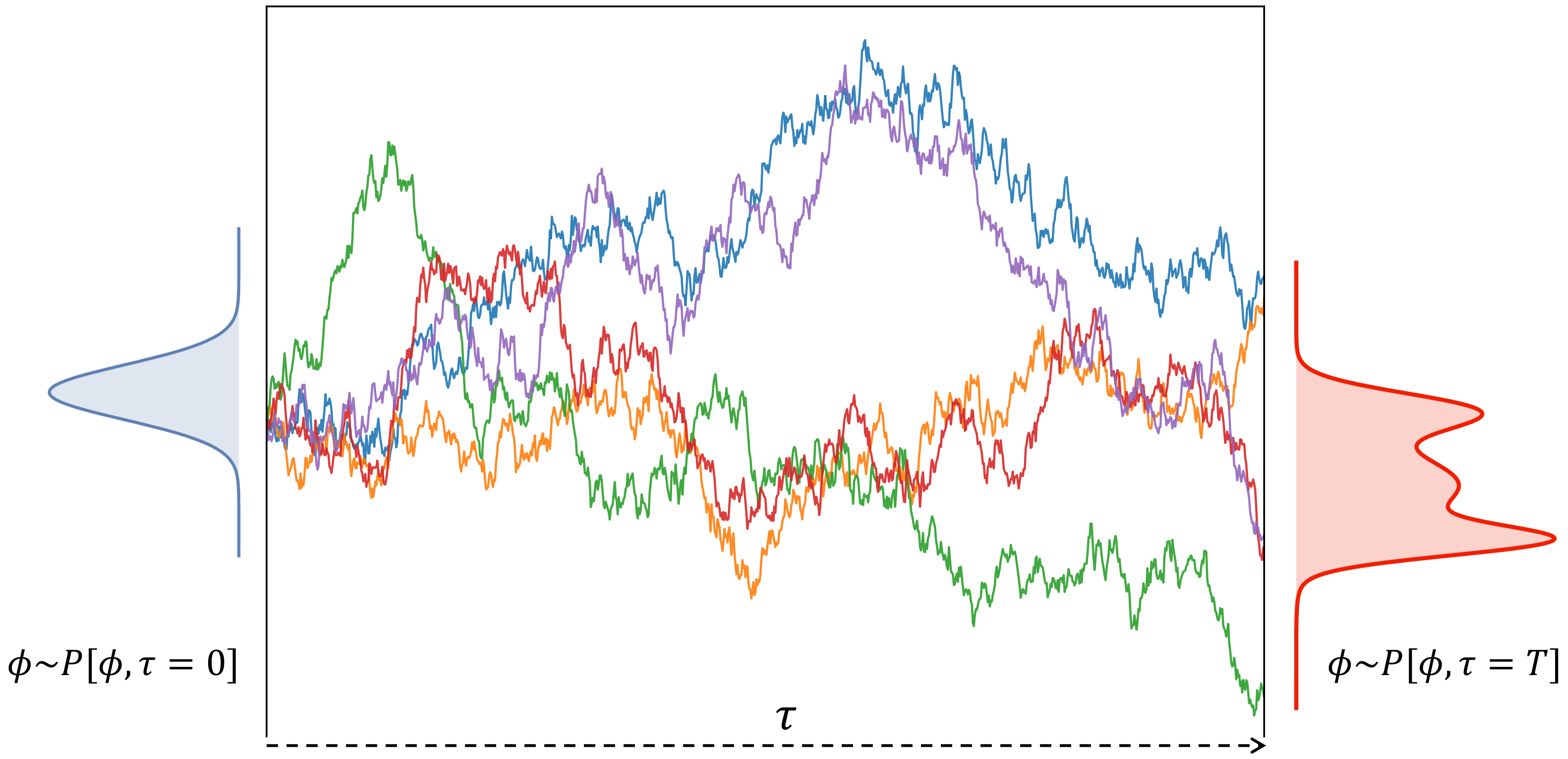
$$\frac{\partial P[\phi,\tau]}{\partial \tau} = \alpha \int d^n x \left\{ \frac{\delta}{\delta \phi} \left(\frac{\delta}{\delta \phi} + \frac{\delta S_E}{\delta \phi} \right) \right\} P[\phi,\tau]$$

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**

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

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

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

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

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

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

DM for Scalar Field

JHEP 05(2024)060

- **Euclidean action** on lattice

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

- Dimensionless form

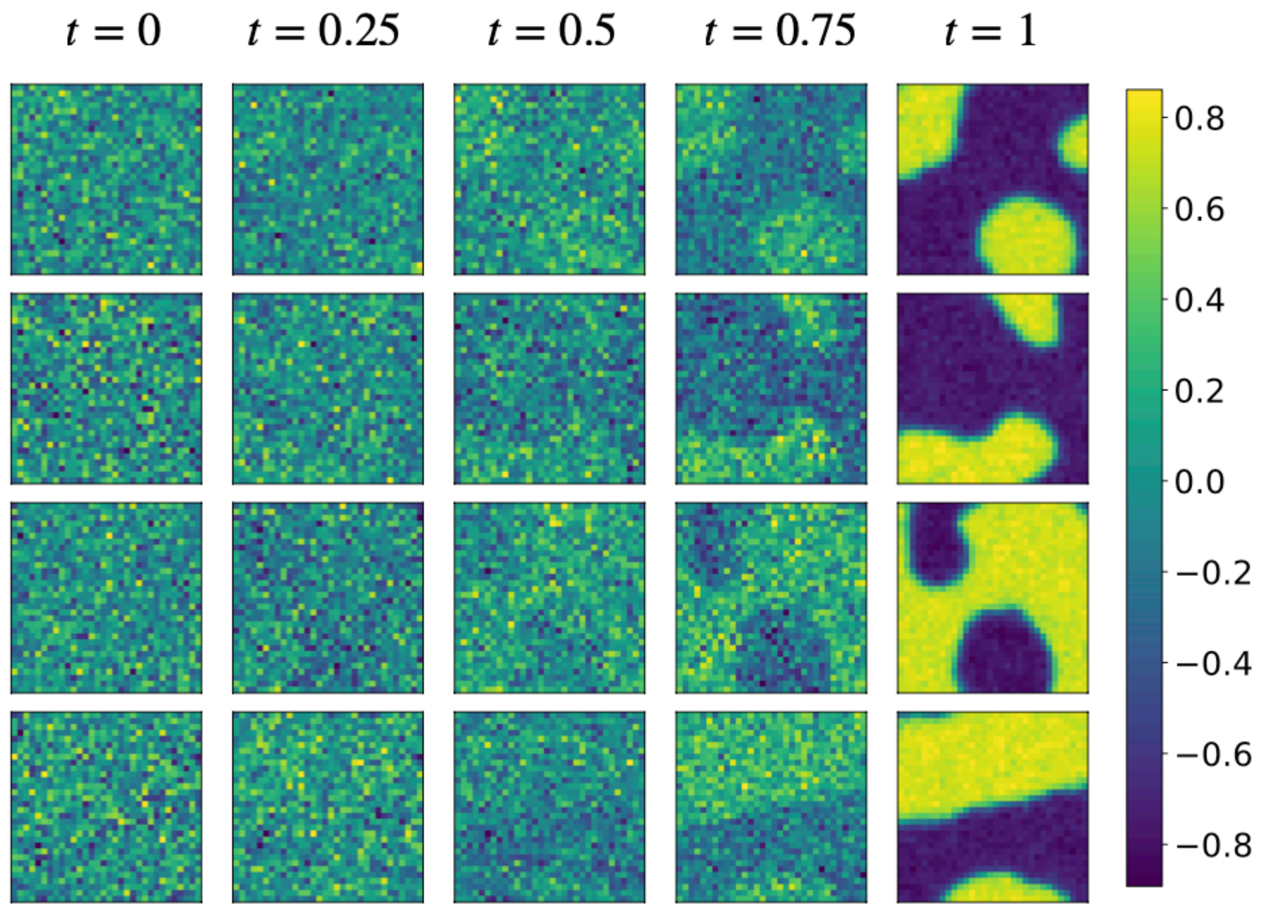
$$S_E = \sum_x \Big[-2\kappa \sum_{\mu=1}^{d} \phi(x)\phi(x+\hat{\mu}) + (1-2\lambda)\phi(x)^2 + \lambda\phi(x)^4 \Big]$$

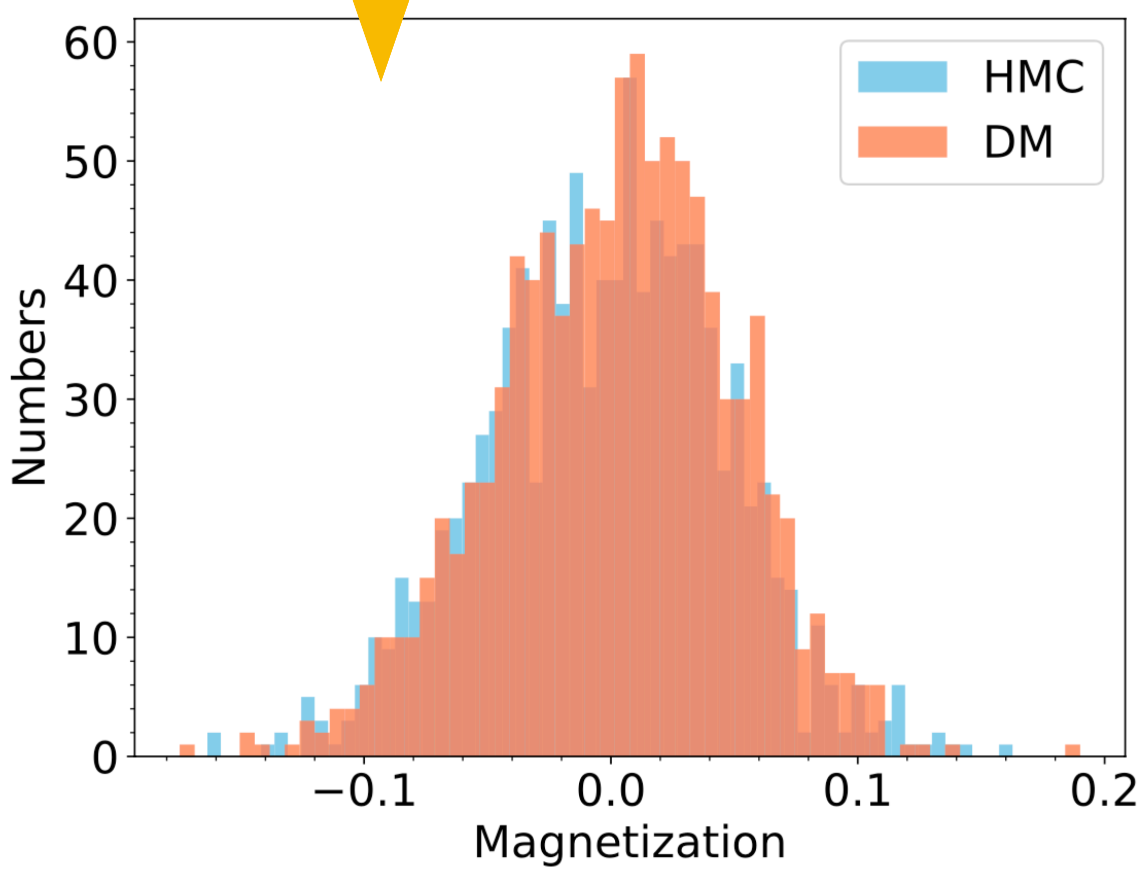
$$a^{\frac{d-2}{2}}\phi_0 = (2\kappa)^{1/2}\phi$$

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

  - 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: $\kappa = 0.21, \lambda = 0.022$

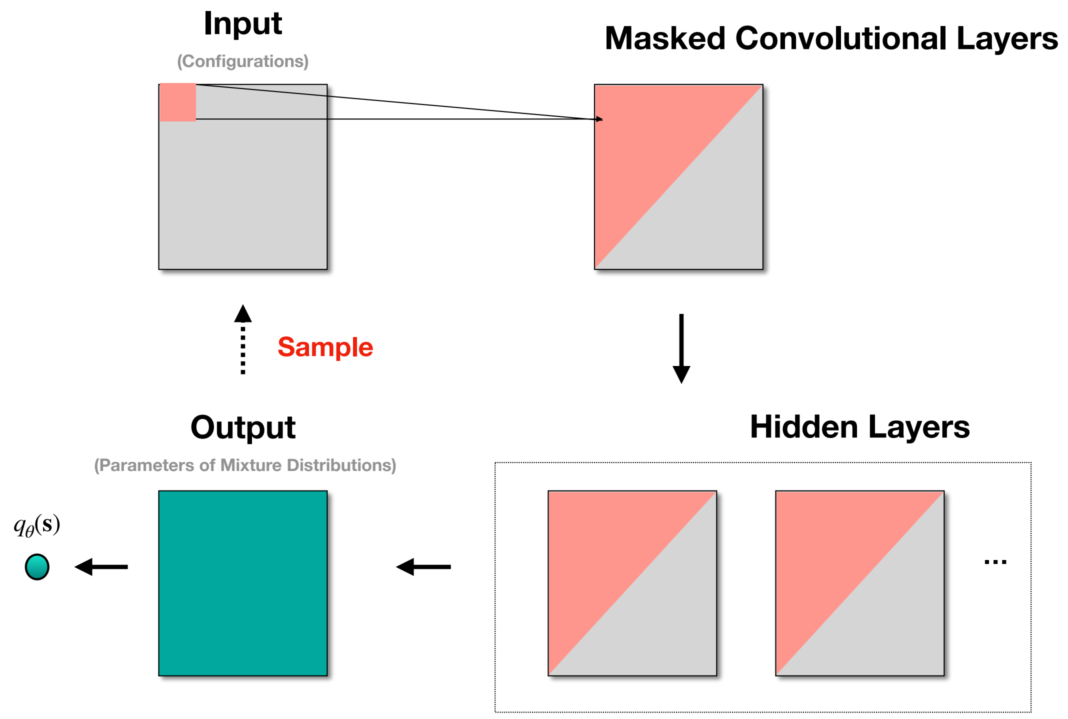




data-set	$\langle M \rangle$	χ_2	U_L
Training(HMC)	0.0012± 0.0007	2.5160 ± 0.0457	0.1042 ± 0.0367
Testing(HMC)	0.0018 ± 0.0015	2.4463 ± 0.1099	-0.0198 ± 0.1035
Generated(DM)	0.0017± 0.0015	2.4227 ± 0.1035	0.0484 ± 0.0959

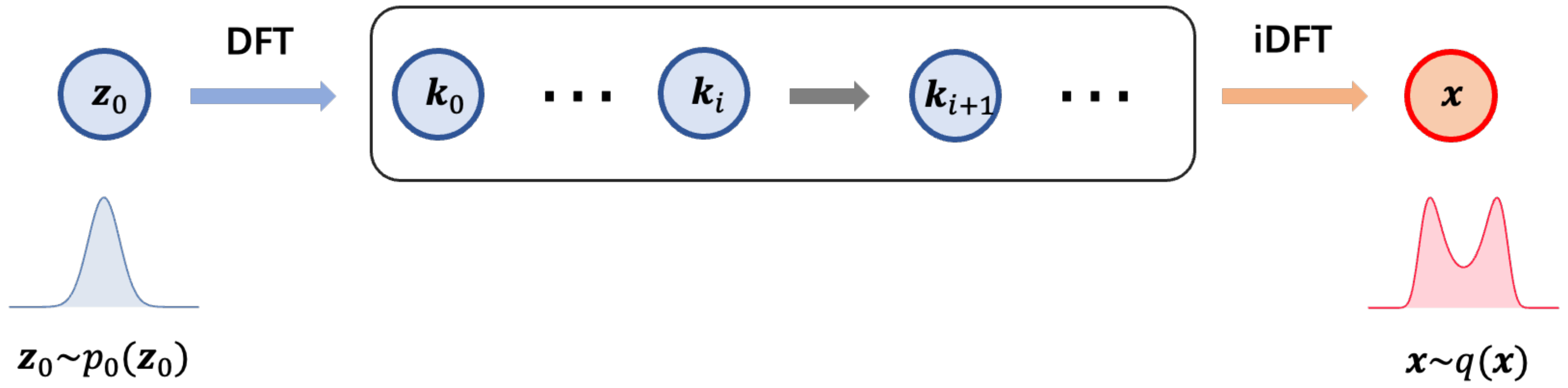
Summary II

- **Generative Models**
  - Probabilistic Models
  - Score Matching
  - Diffusion Models
    - **Stochastic Quantization** scheme
- **Future works**
  - **Diffusion models as SQ**
  - 2+1D **Gauge Field**
  - Complex Langevin Method(CLM) for **Fermions**

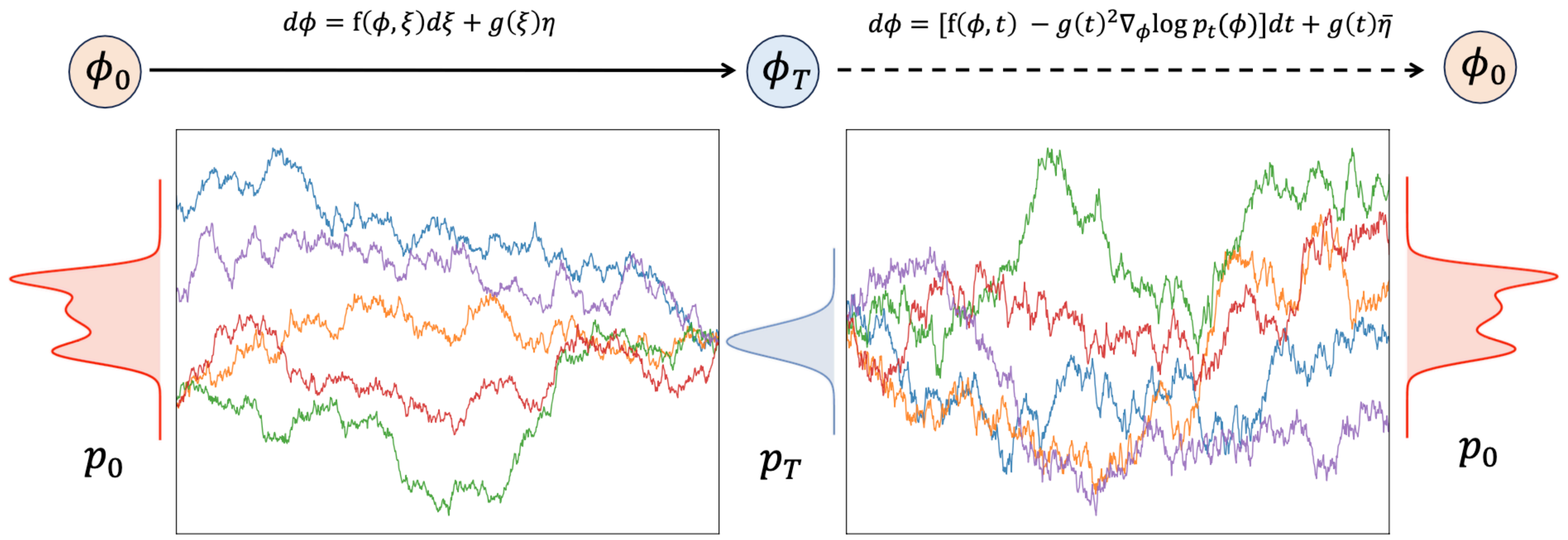


Chinese Phys. Lett. 39, 120502 (2022)

Phys. Rev. D 107, 056001



JHEP 05(2024)060



Thank You !

ML meets Physics, Opportunities and Challenges

Kouzou Sakai @Quanta Magazine

Representation Learning

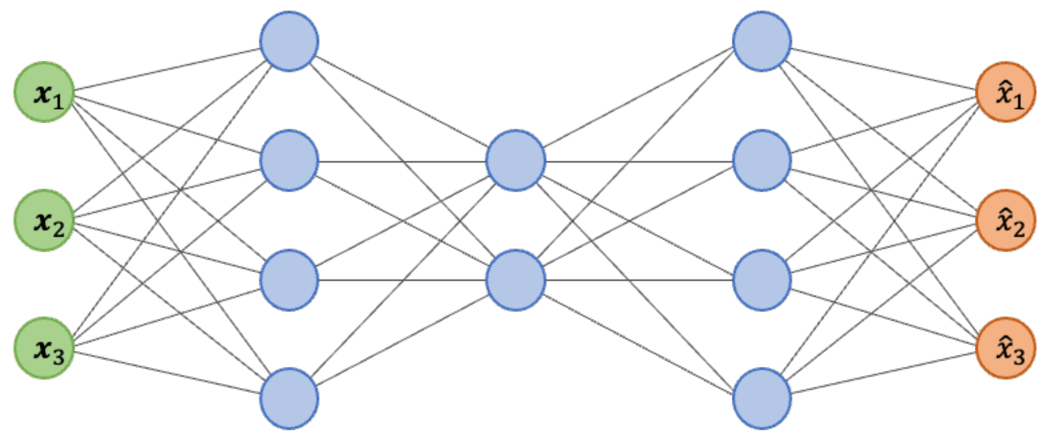
$$g_\theta : X \rightarrow Y$$

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

Forward Mapping, f_θ

Physics

Model Parameters/ Properties/States



AutoEncoder

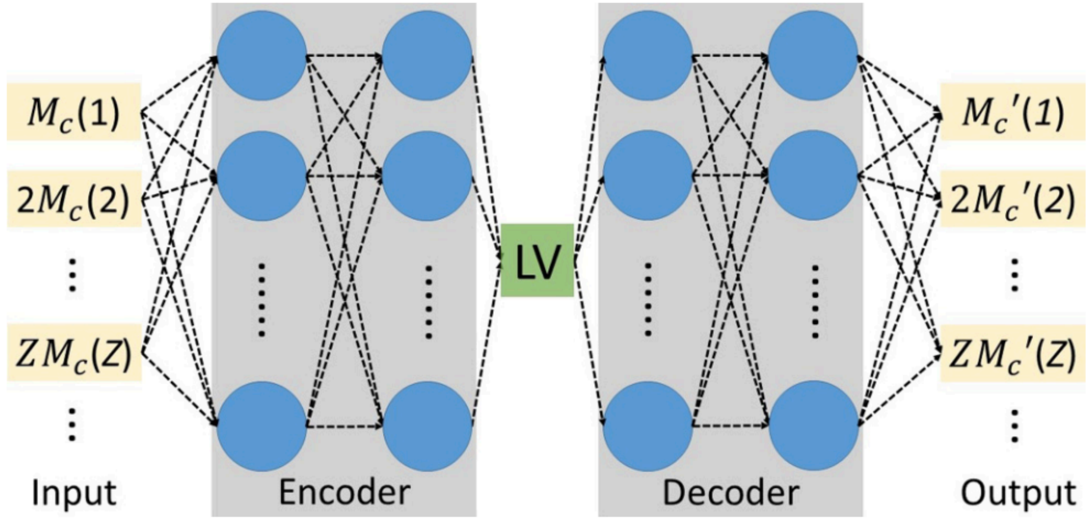
Data

Observations

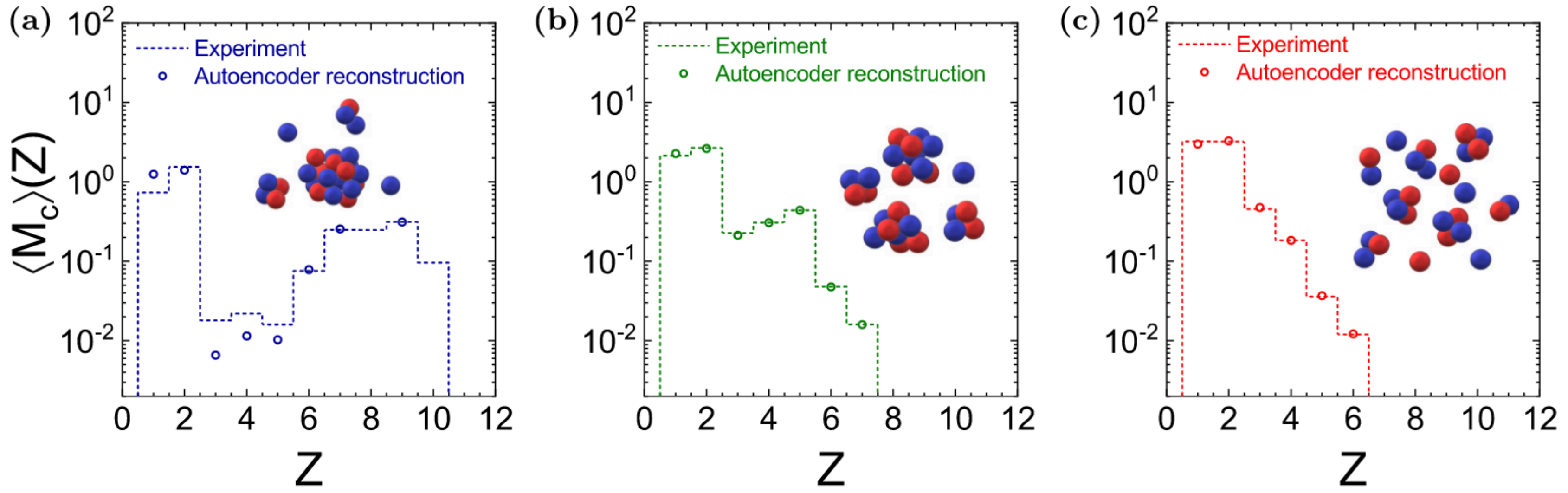
Inverse Mapping, g_θ

Representation Learning

Recognizing nuclear liquid-gas phase transition



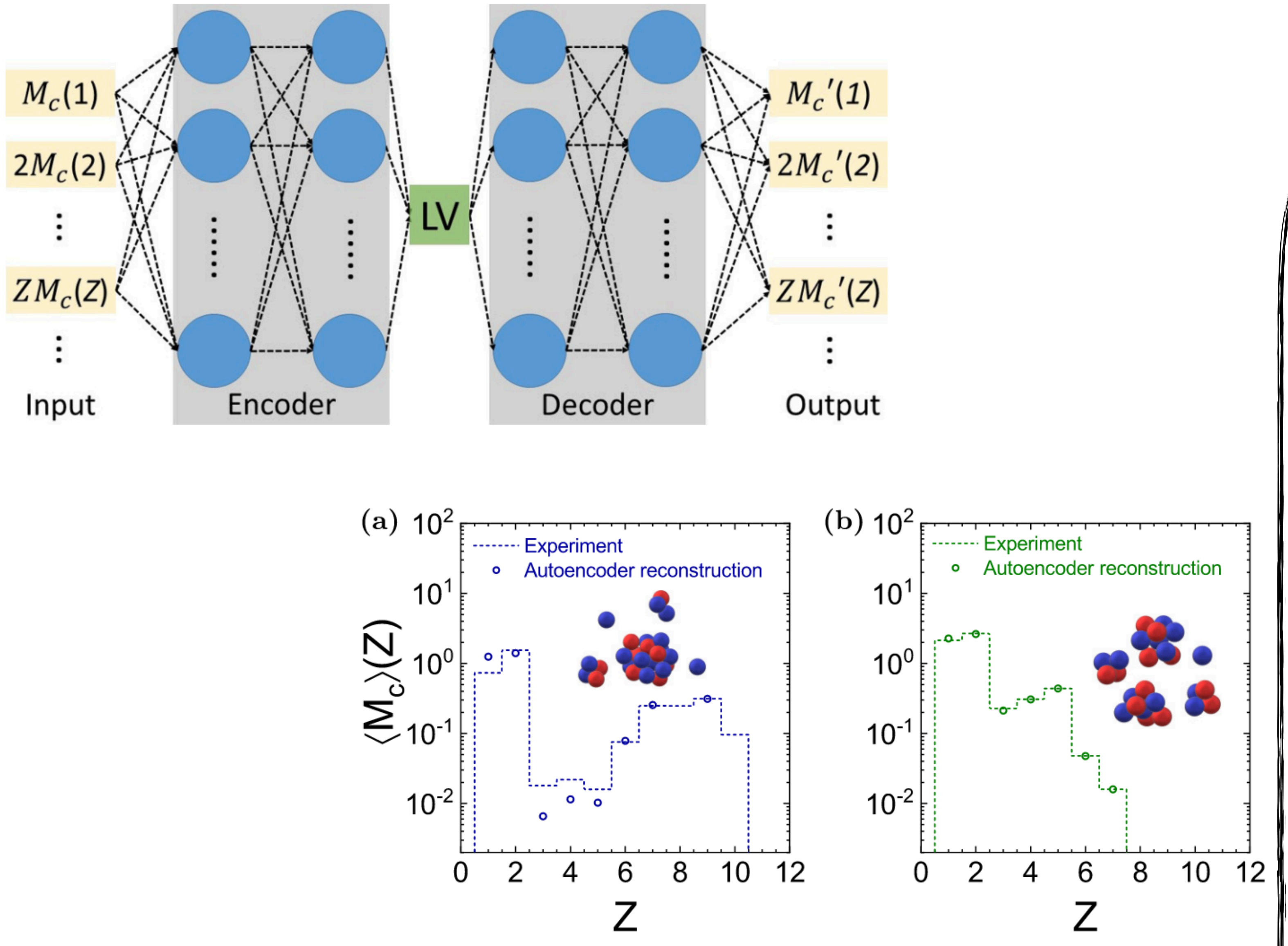
Anomaly Detection
Latent Variable Extraction



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)

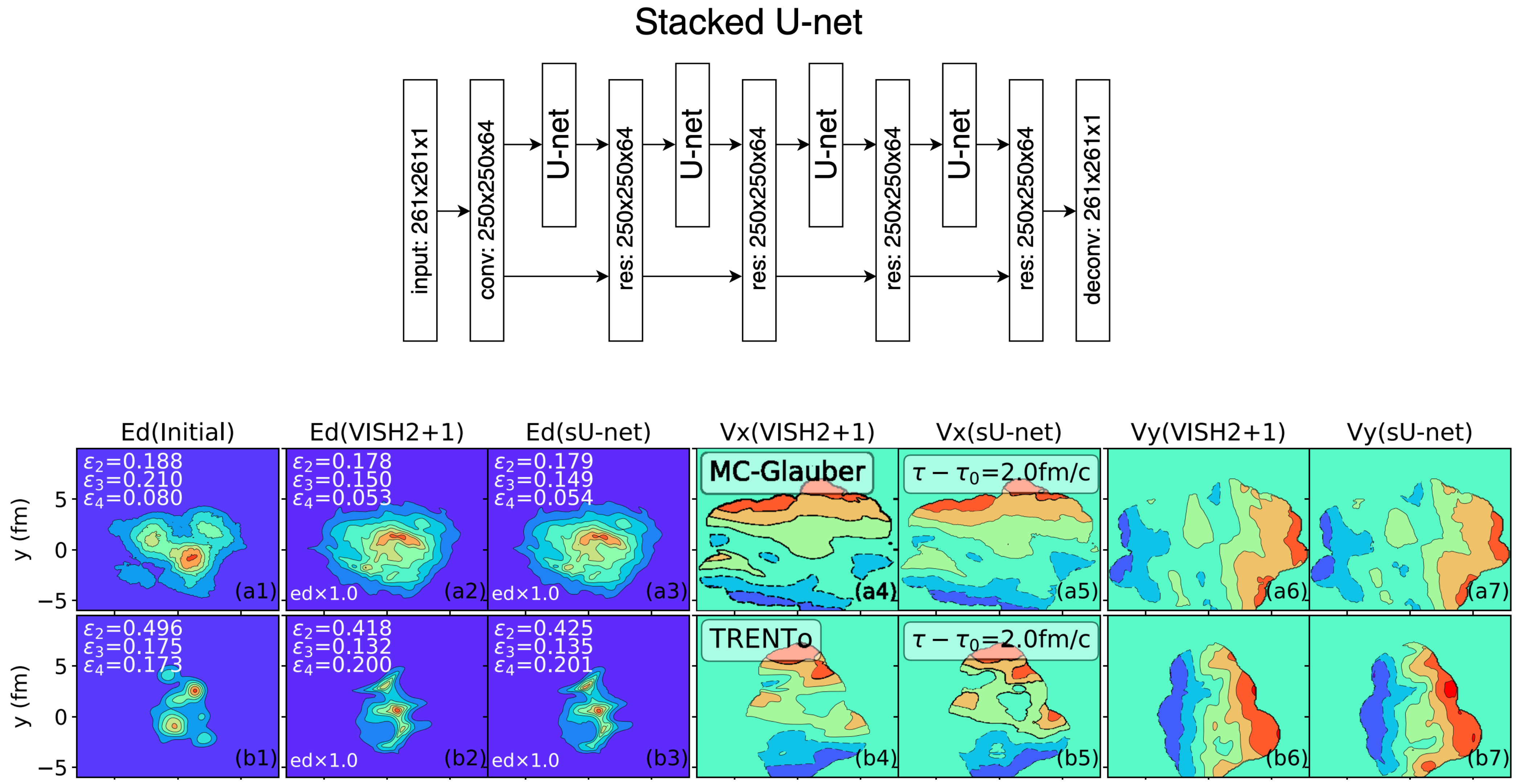
Representation Learning

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

1. Building Neutron Star EoS

Phys. Rev. D 107, 083028

Uncertainty estimations

- x : reconstructed EoSs given a sample
- $O(x)$: observables, M, R, P

Variance $\sigma(O)^2 = \langle \hat{O}^2 \rangle - \bar{O}^2$

Mean $\bar{O} = \langle \hat{O} \rangle = \sum_{j}^{N} w^{(j)} O^{(j)}$

Recall: Importance Sampling

- x : random variables
- $f(x)$: observables
- n : number of samples
- $p(x)$: original(true) distribution
- $q(x)$: reference distribution

$$E[f(x)] = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$

Weights

$$w^{(j)} = \frac{\text{Posterior}(\boldsymbol{\theta}_{\text{EoS}}^{(j)} \,|\, \text{data})}{\text{Proposal}(\boldsymbol{\theta}_{\text{EoS}}^{(j)})} \propto \frac{\boxed{p(\text{data} \,|\, \boldsymbol{\theta}_{\text{EoS}}^{(j)})} \cdot \cancel{\text{Prior}(\boldsymbol{\theta}_{\text{EoS}}^{(j)})}}{\boxed{p(\boldsymbol{\theta}_{\text{EoS}}^{(j)} \,|\, \text{samples}^{(j)})} \cdot \boxed{p(\text{samples}^{(j)} \,|\, \text{data})} \cdot \cancel{\text{Prior}(\text{data})}}$$

L2 : $exp(-\lambda||\theta||_2^2) \sim 1, \quad \lambda = 10^{-8}$

$$p(\text{data} \,|\, \boldsymbol{\theta}_{\text{EoS}}^{(j)}) \propto \exp(-\chi^2(M_{\boldsymbol{\theta}_{\text{EoS}}^{(j)}}, R_{\boldsymbol{\theta}_{\text{EoS}}^{(j)}}))$$

$$\tilde{w}^{(j)} = w^{(j)} / \sum_j w^{(j)}$$

normalization

$$p(\boldsymbol{\theta}_{\text{EoS}}^{(j)} \,|\, \text{samples}^{(j)}) = 1$$

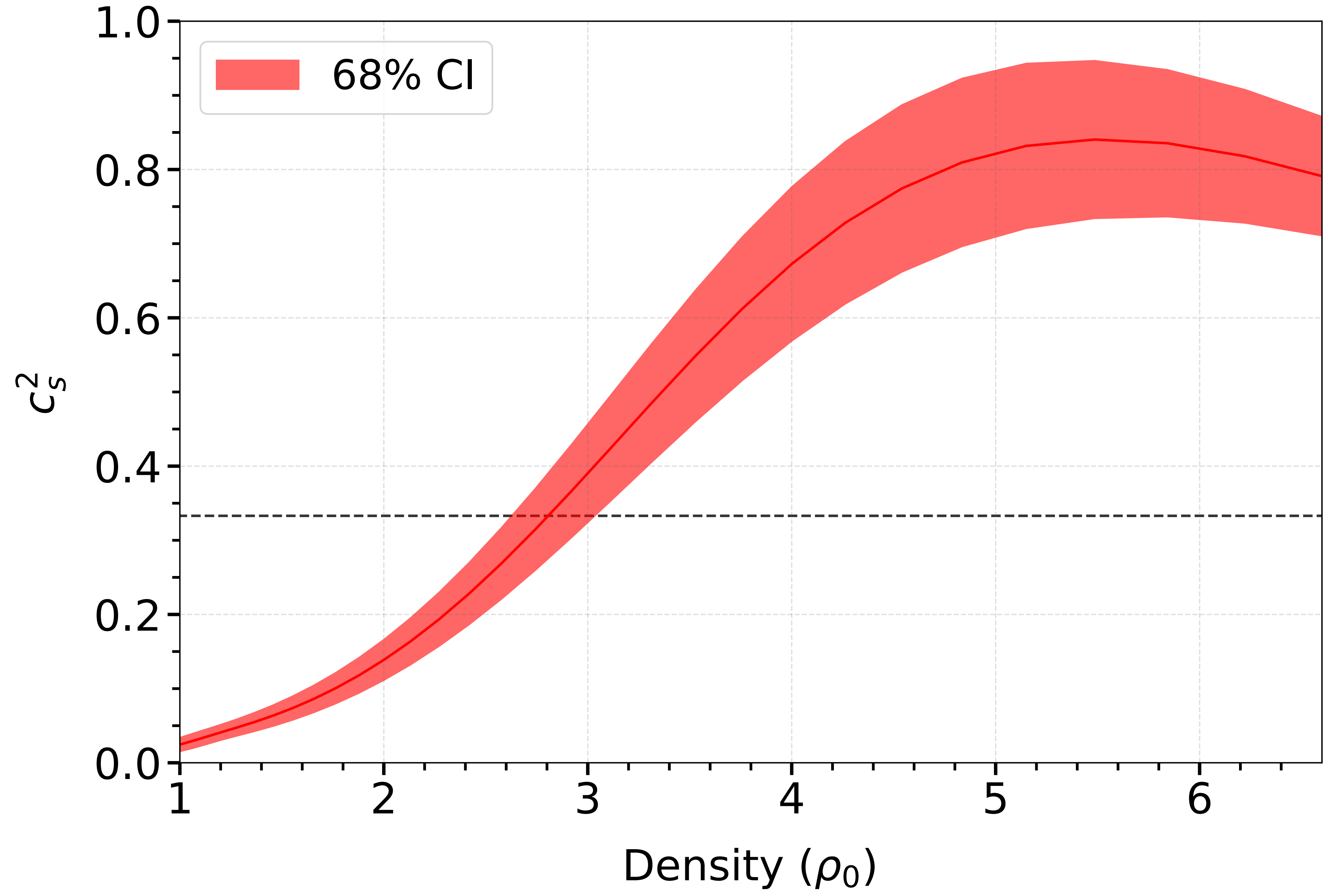
deterministic

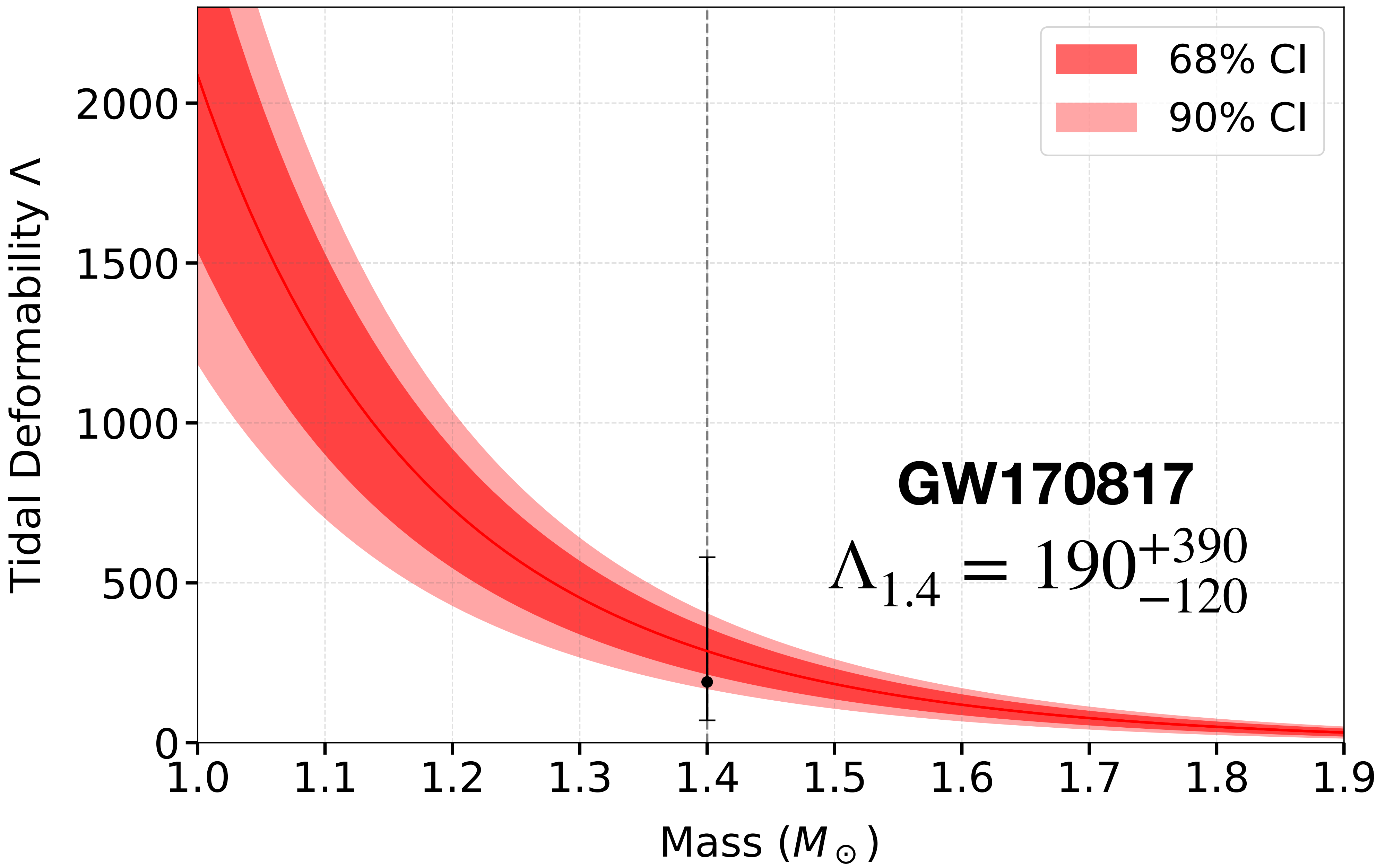
$$p(\text{samples} \,|\, \text{data}) = \mathcal{N}(M_{\text{obs}}, \Delta M^2)\mathcal{N}(R_{\text{obs}}, \Delta R^2)$$

Backups I

1. Building Neutron Star EoS

Phys. Rev. D 107, 083028





Our results, $\tilde{\Lambda}_{1.4} = 286.47^{+115.9}_{-115.9}$

Backups I

2. 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

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

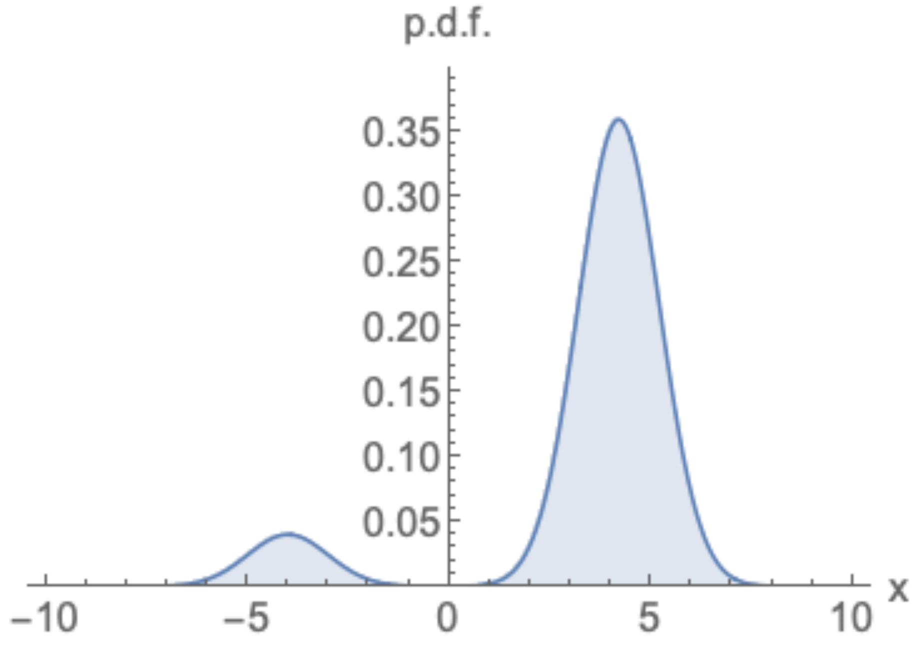
J. Phys. A: Math. Gen., Vol. 11, No. 9, 1978. Printed in Great Britain.

Backups II: Diffusion Models

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

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

Maximum Likelihood Estimation



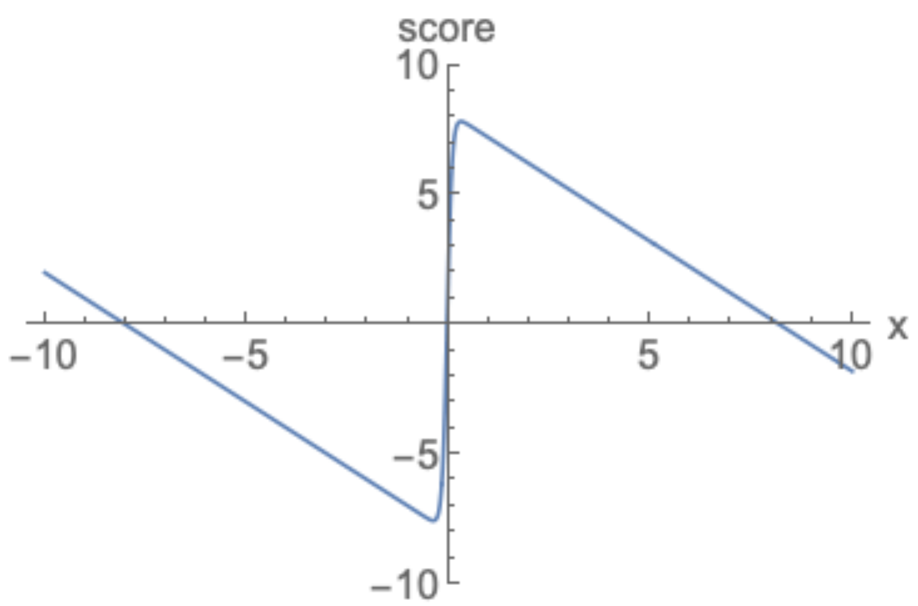
Parameterizing probability density functions

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

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

Approaching Score Function

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



Parameterizing score functions

@Yang Song's Blog

Backups II: Diffusion Models

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

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

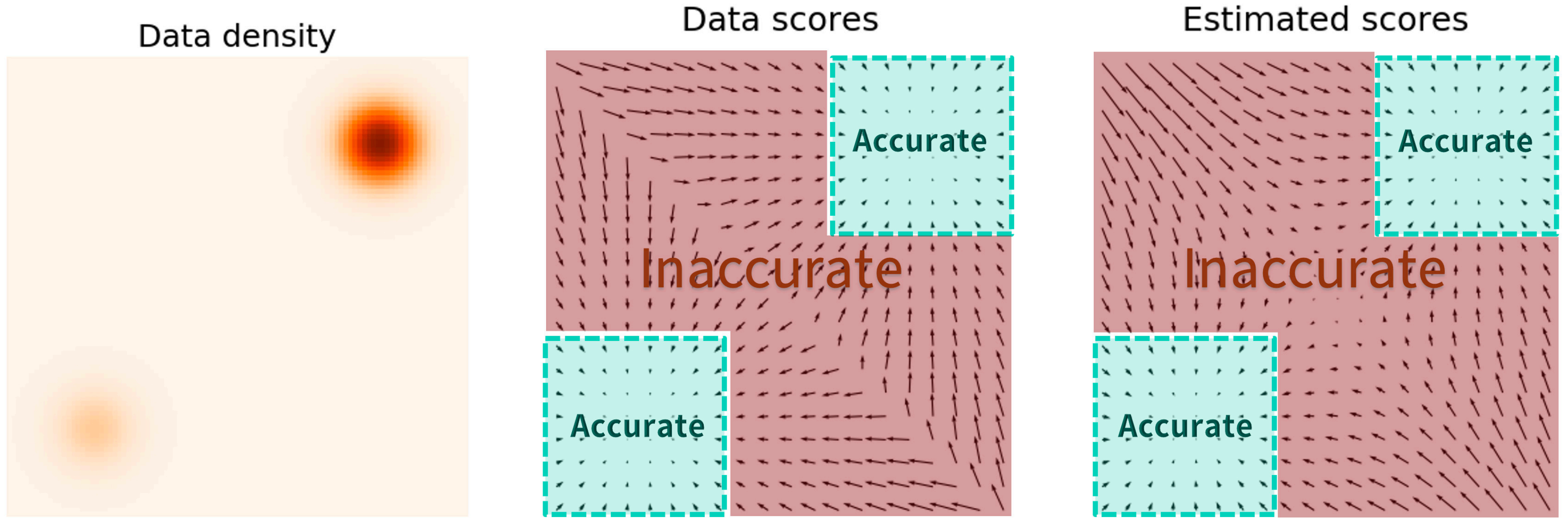
$$p_\sigma(\tilde{x}|x) \sim N(\tilde{x}; x, \sigma^2 I), \epsilon \sim N(0, I)$$

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

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...

🤔

Backups II: 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

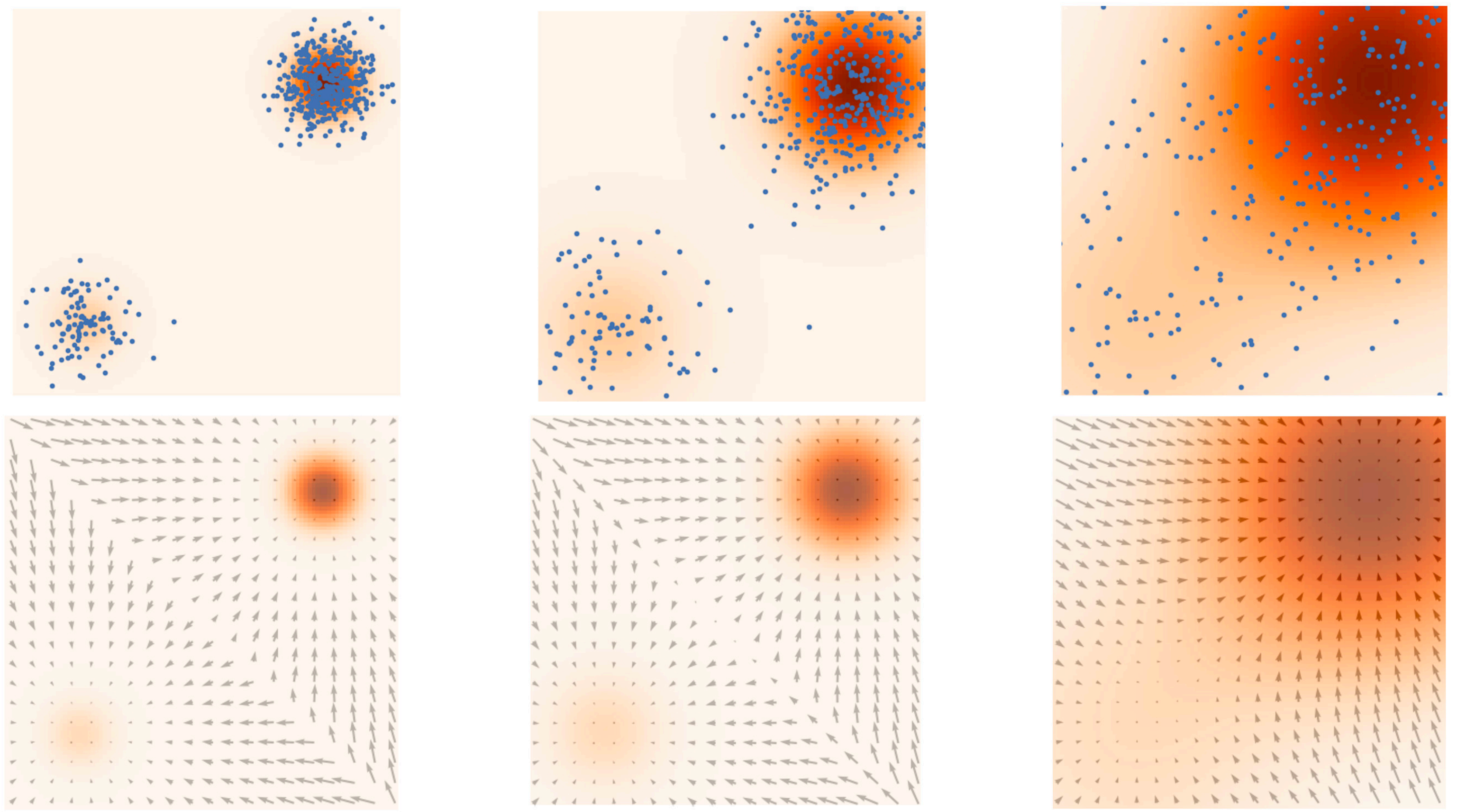
Small noise → Good approximation to data, poor in low-density region!
Large noise → Poor approximation to data, good in low-density region!

Multiple noise perturbations!

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

Matching multi-noise perturbed distributions

$\sigma_1 \quad < \quad \sigma_2 \quad < \quad \sigma_3$



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

- 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.

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

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

Backups II: 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

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

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

$$\boxed{\begin{aligned} \frac{d\mu}{dt} &= 0 \\ \frac{d\sigma^2}{dt} &= \sigma^{2t} \end{aligned}}$$

$t \in [0, \tau]$

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

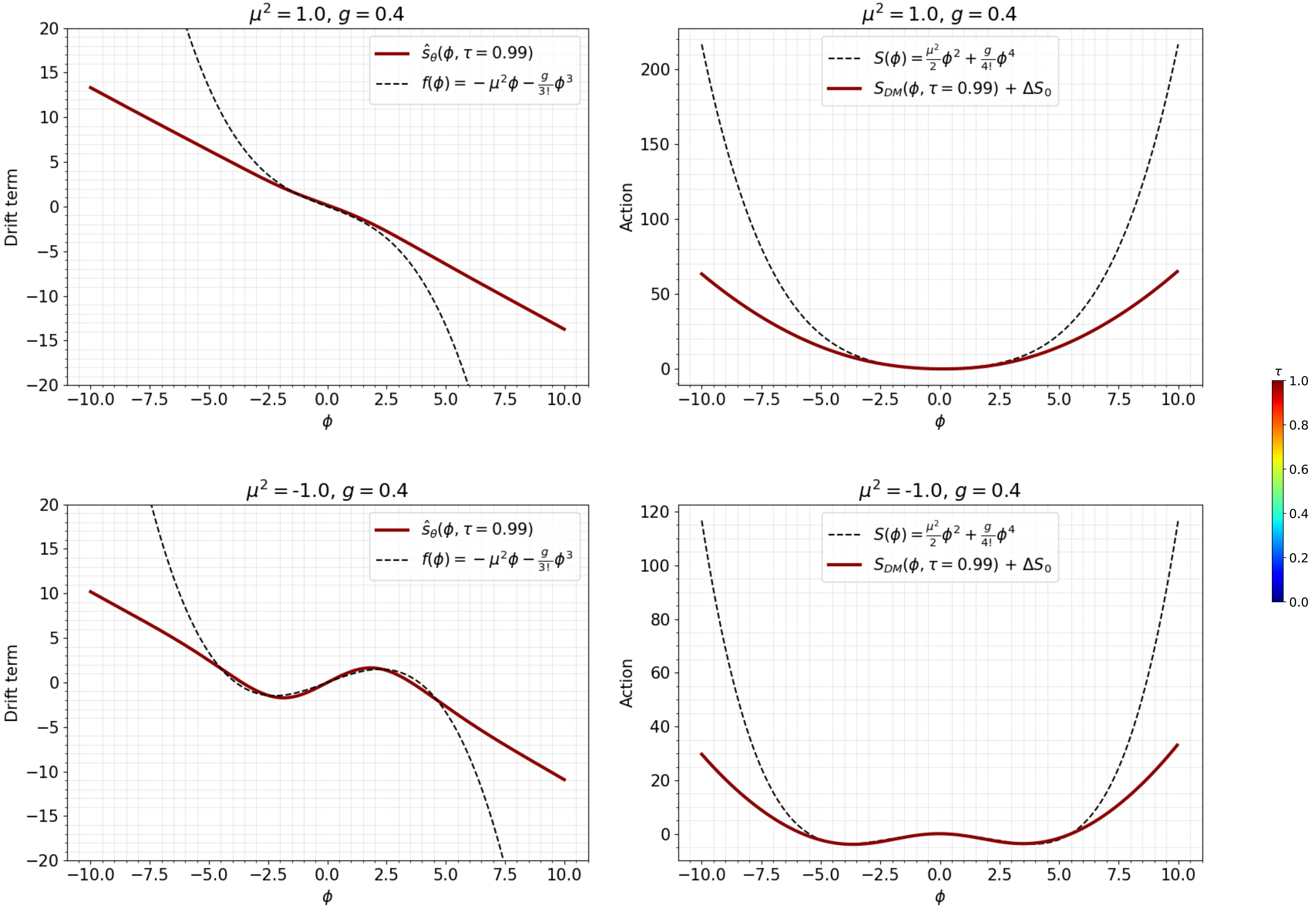
- **Choose the noise scheme as, $\sigma_i \equiv \sigma^\tau$, where τ indicates "time-step" for adding noise.**

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

Training is Matching

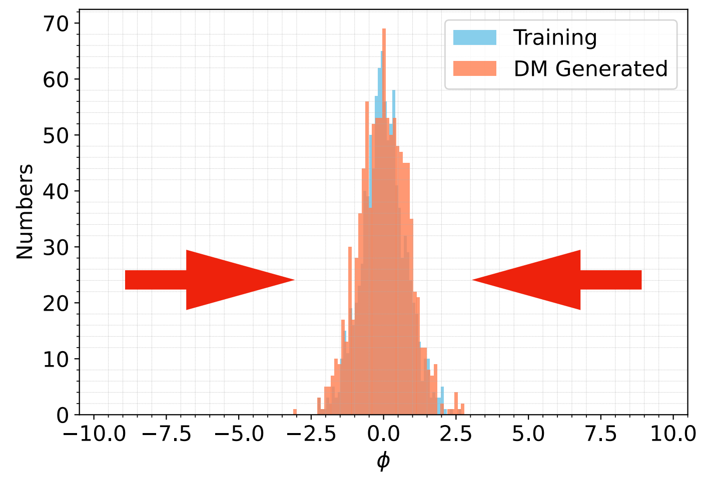
Backups II: Diffusion Models

JHEP 05(2024)060

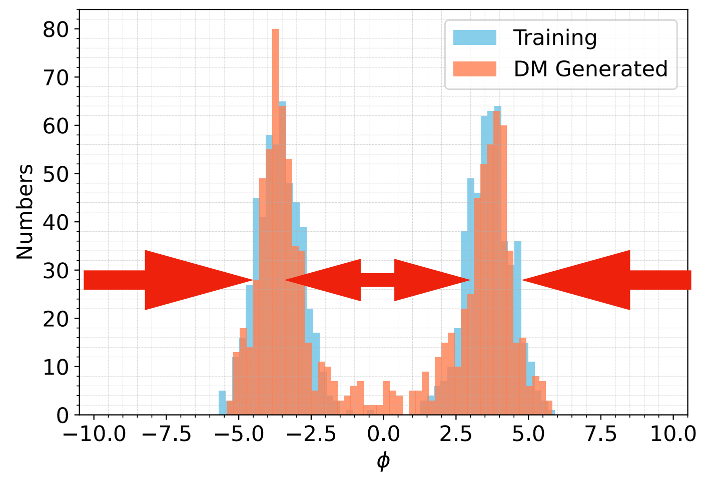


$$S(\phi) = \frac{\mu^2}{2}\phi^2 + \frac{g}{4!}\phi^4,$$

$$f(\phi) = -\mu^2\phi - \frac{g}{3!}\phi^3$$



0+0 dimensional FT (1 variable)

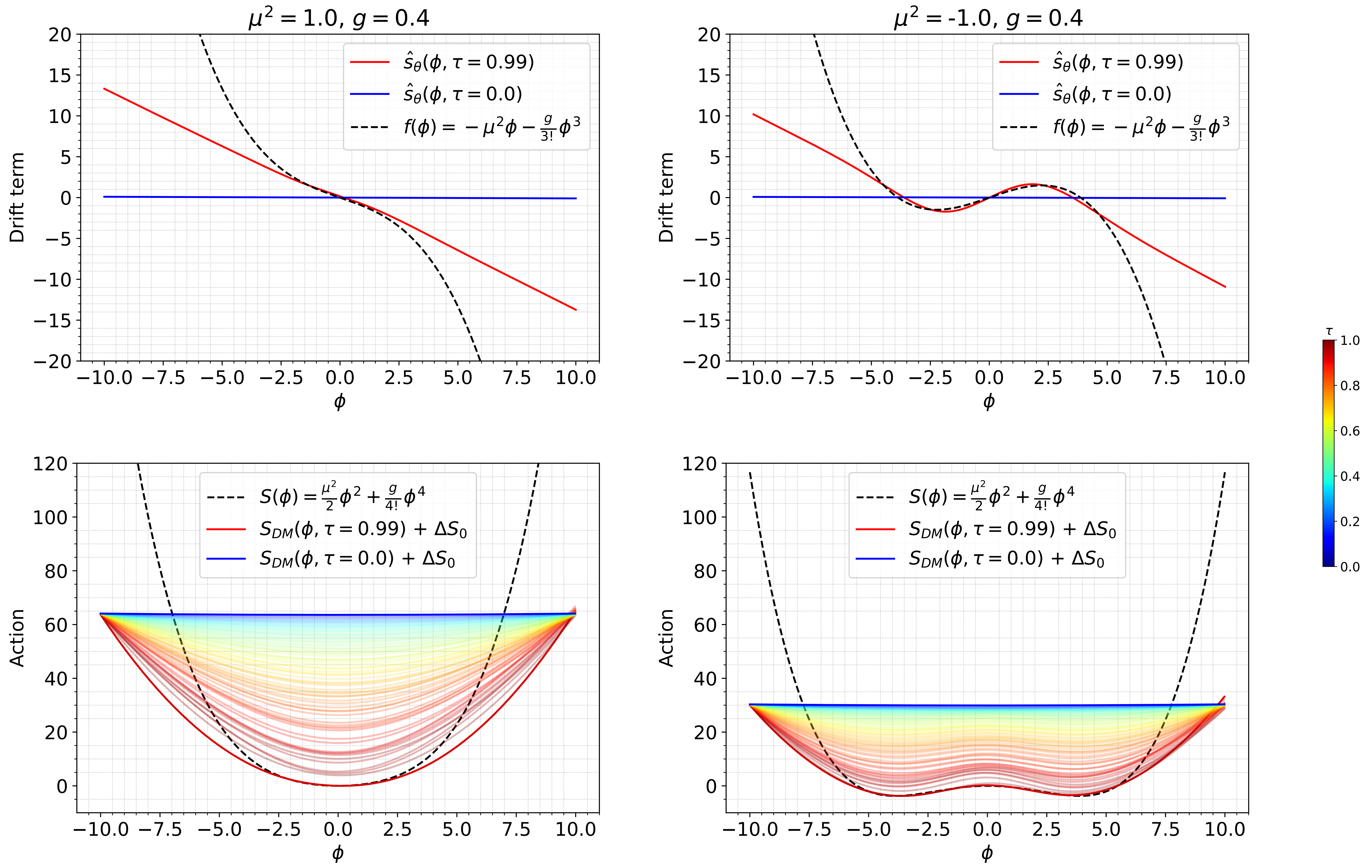


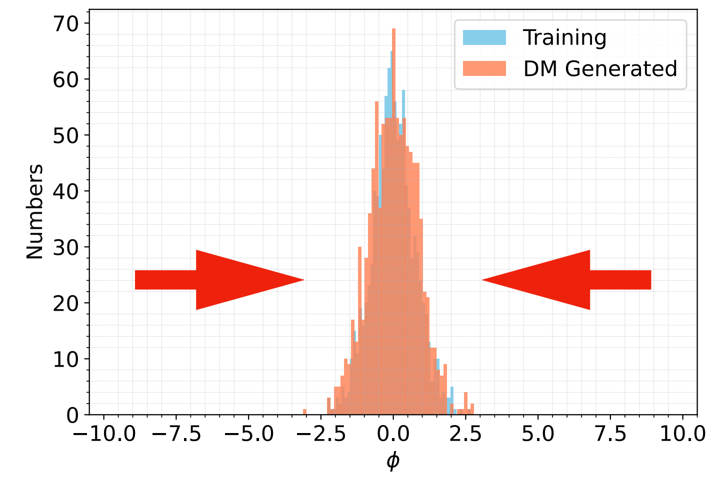
Backups II: Diffusion Models

JHEP 05(2024)060

$$S(\phi) = \frac{\mu^2}{2}\phi^2 + \frac{g}{4!}\phi^4,$$

$$f(\phi) = -\mu^2\phi - \frac{g}{3!}\phi^3$$





0+0 dimensional FT
(1 variable)