NUSYS 2024 @ ZhuHai

# Deep Learning for high energy nuclear physics


Long-Gang Pang 庞龙刚

华中师范大学粒子物理研究所

夸克与轻子物理教育部重点实验室

# What is deep learning

Yann LeCun

**Deep learning is constructing networks of parameterized functional modules & training them from examples using gradient-based optimization**

# DL: Neural Network with multi hidden layers

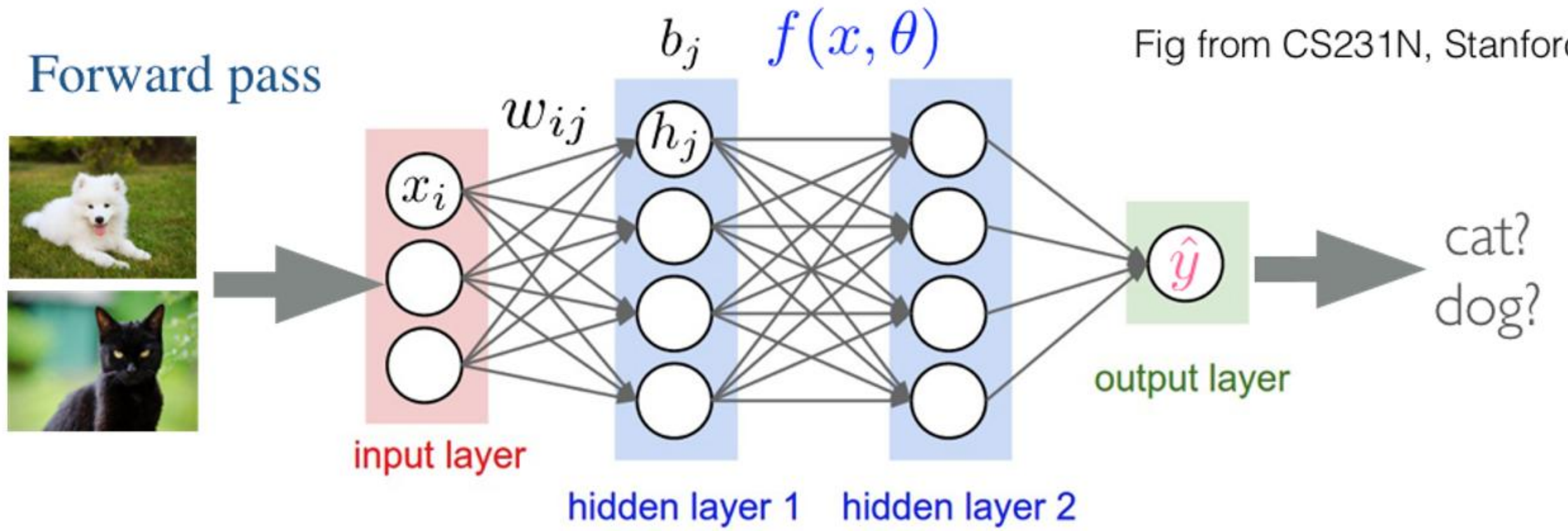



Fig from CS231N, Stanford

## Linear operation

$$z_j = \sum_{i=1}^{N} x_i w_{ij} + b_j$$

scaling, rotating, boosting, changing dimensions

## Non-linear activation function $h_j = \sigma(z_j)$

(a) Sigmoid

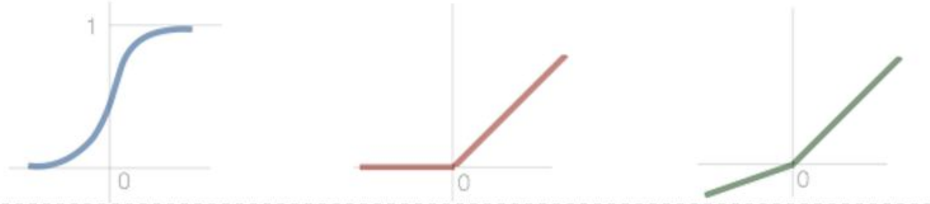
$$\sigma(z) = \frac{1}{1+\exp(-z)}$$

(b) ReLU

$$\sigma(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}$$

(c) PReLU

$$\sigma(z) = \begin{cases} z, & z > 0 \\ az, & z \leq 0 \end{cases}$$



# How does the network learn

Input x → Neural network $\hat{y} = f(x, \theta)$ → $\hat{y}$: Network prediction, $y$: True label

Loss(cost) function

$$L = \frac{1}{N}\sum_i (\hat{y}_i - y_i)^2$$

Gradient decent

$$\theta = \theta - \epsilon \frac{\partial L}{\partial \theta}$$

Back propagation

1. The network parameters are initialized with random numbers and the network make random guess
2. Compute the loss function L which is a functional of the network
3. Use auto-diff to compute the negative gradients of network parameters
4. Use SGD-like algorithms to update network parameters

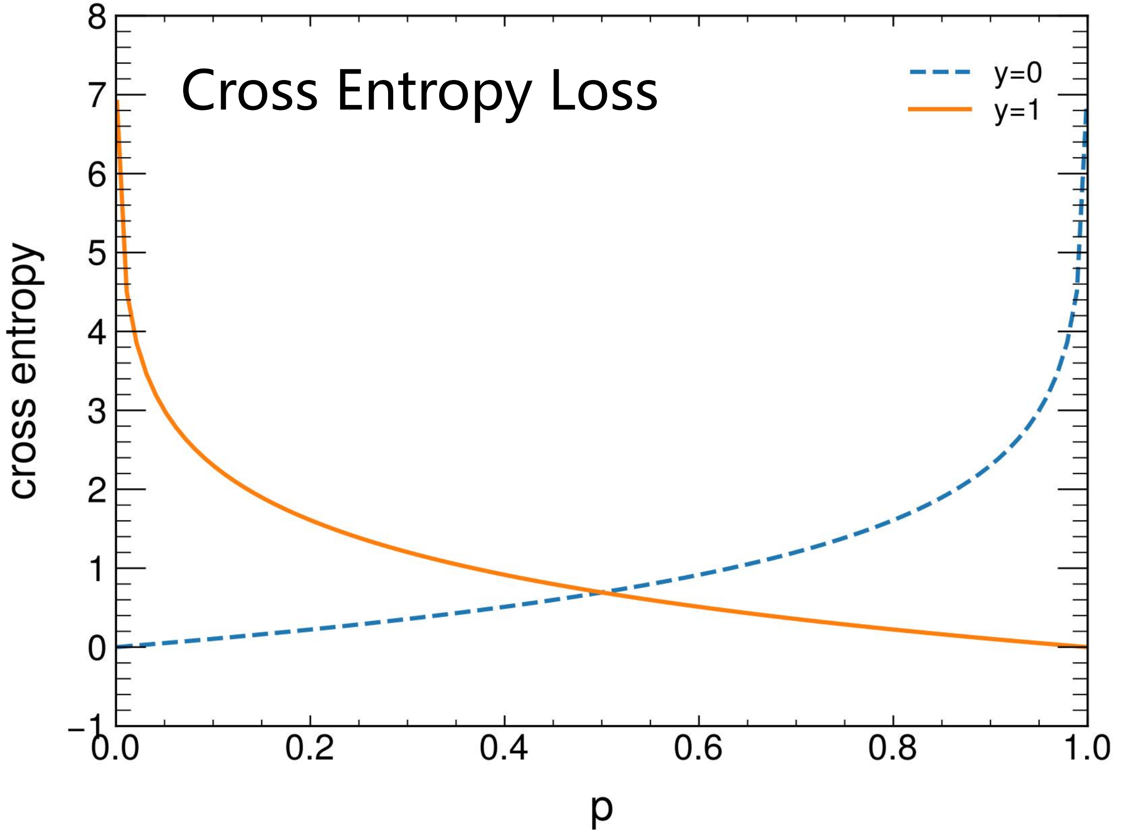
# Loss functions

$$\mathrm{KL}(Q||P) = \sum_{i=1}^{n} Q(x_i)\log\frac{Q(x_i)}{P(x_i,\theta)}$$

$$= \sum_{i=1}^{n}[Q(x_i)\log Q(x_i) - Q(x_i)\log P(x_i,\theta)]$$

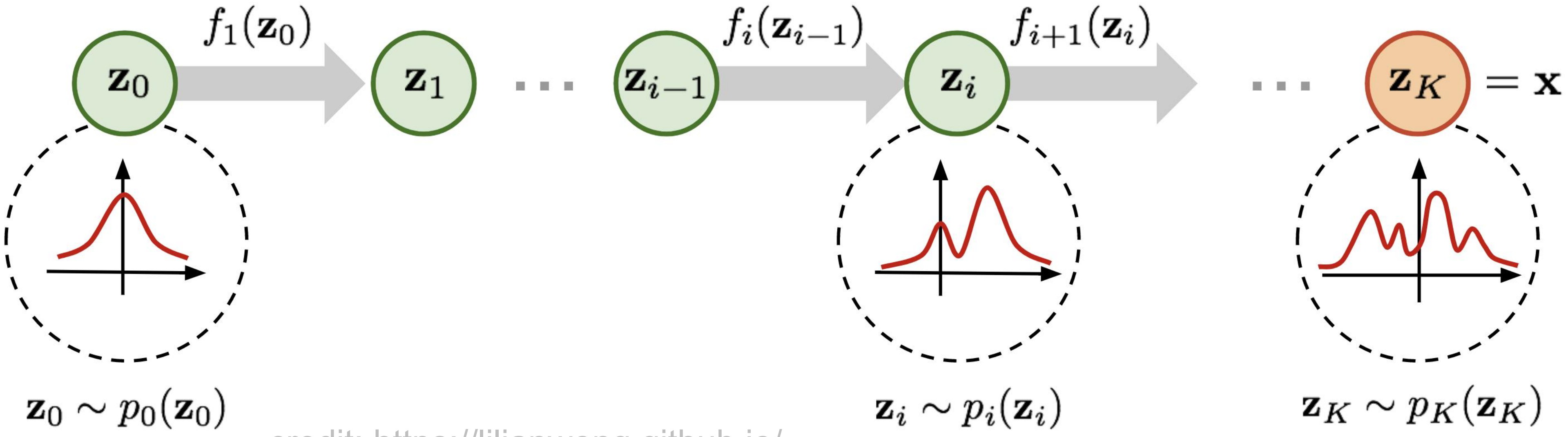
**Cross Entropy:** $$L(q,p) = -\sum_{i=1}^{n} q(x_i)\log p(x_i,\theta)$$

**Negative Log Likelihood:** $$L = -\sum_{i=1}^{n}\log p(x_i,\theta)$$




Cross Entropy Loss punishes wrong and confident predictions

# NLL loss in flow model




credit: https://lilianweng.github.io/

**Normalizing flow:**

$$\mathbf{z}_{i-1} \sim p_{i-1}\left(\mathbf{z}_{i-1}\right)$$

$$\mathbf{z}_i = f_i\left(\mathbf{z}_{i-1}\right), \mathbf{z}_{i-1} = f_i^{-1}\left(\mathbf{z}_i\right)$$

$$p_i\left(\mathbf{z}_i\right) = p_{i-1}\left(f_i^{-1}\left(\mathbf{z}_i\right)\right)\left|\det \frac{d f_i^{-1}}{d \mathbf{z}_i}\right|$$

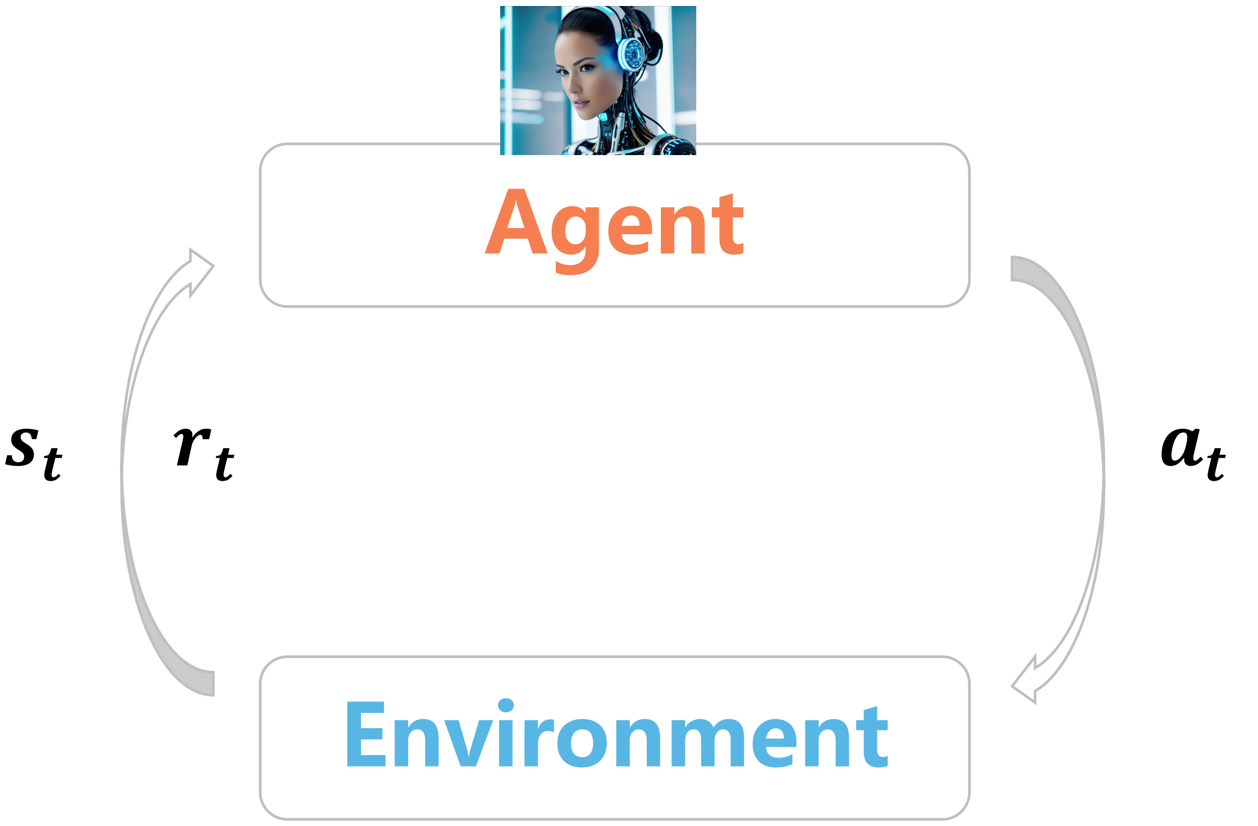
**NLL loss:**

$$\log p(x) = \log p_0\left(\mathbf{z}_0\right) - \sum_{i=1}^{K} \log \left|\det \frac{d f_i}{d \mathbf{z}_{i-1}}\right|$$

$$\text{Loss} = -\log p(x)$$

# NLL loss in RL

Try to find NLL in the policy gradient algorithm, which is quite popular in RL




**Policy gradient:**

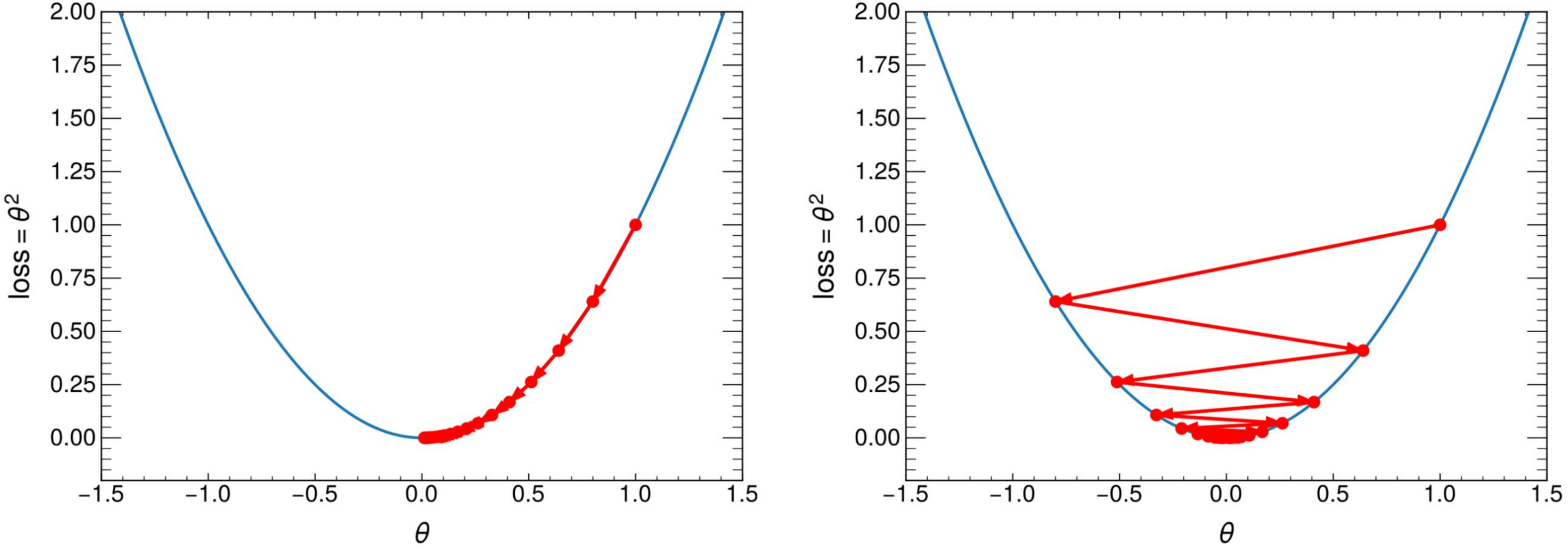
$$
\text{for } t \leftarrow 0 \text{ to } T
$$

$$
\text{do} \begin{cases} G \leftarrow \sum_{k=t}^{T} \gamma^{k-t} \cdot r_k \\ \theta \leftarrow \theta + \alpha \cdot \gamma^{t} \cdot \nabla_\theta \log \pi\left(s_t, a_t; \theta\right) \cdot G \end{cases}
$$

# SGD

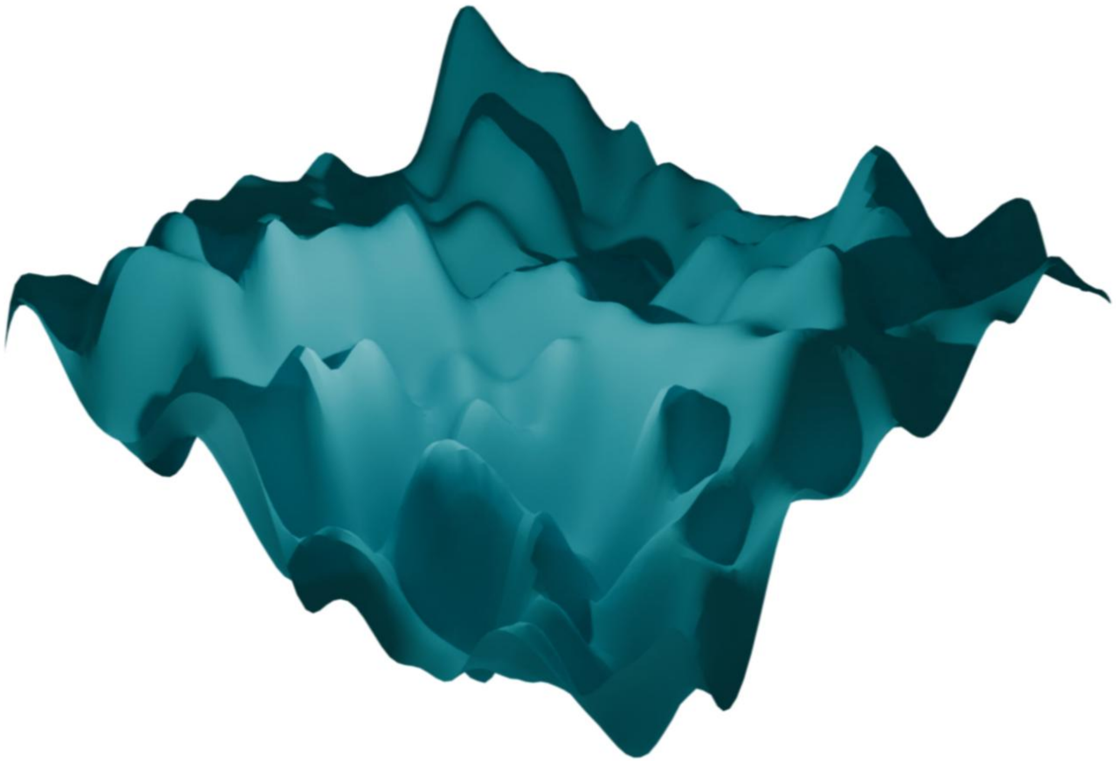
$$\theta = \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta l_i$$

**Example:** $L(\theta) = \theta^2$, simple 1D gradient descent, $\theta = \theta - \alpha \times 2\theta$,



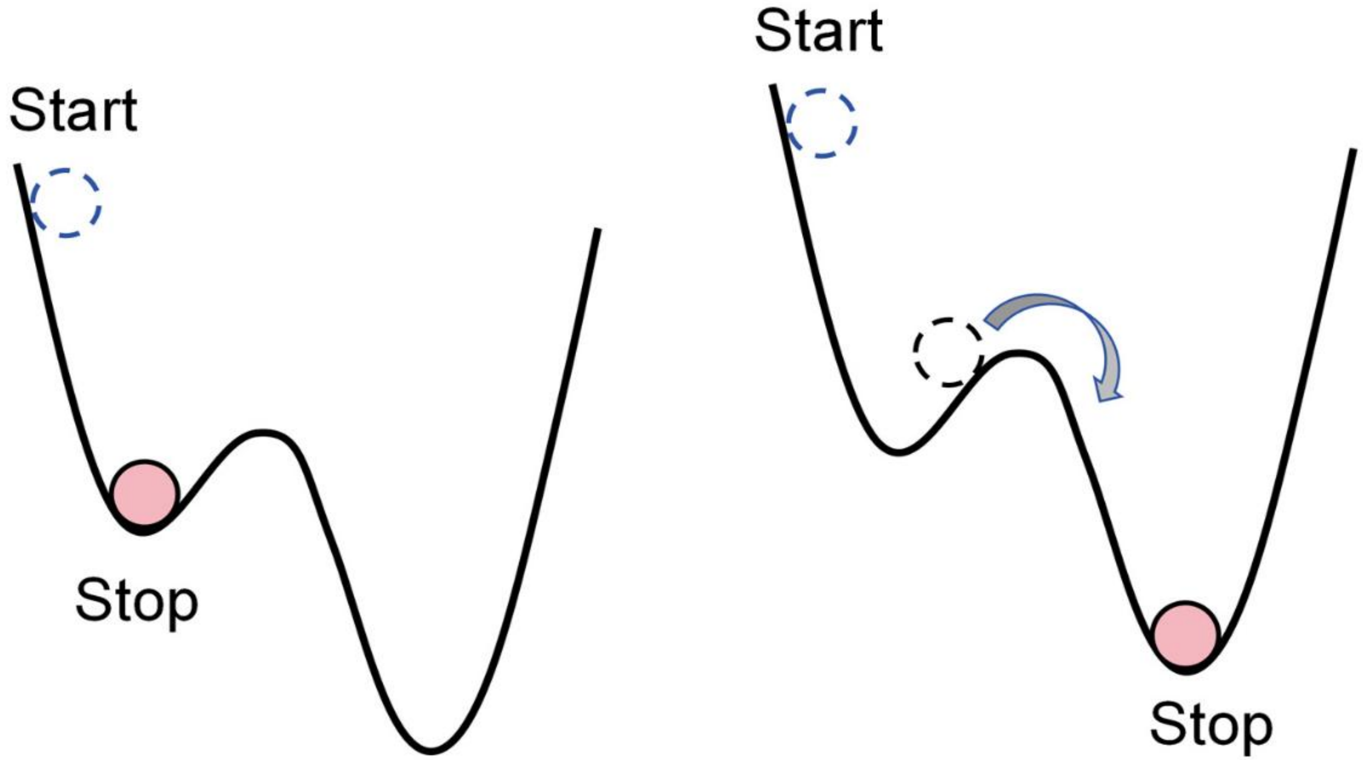

(a) (b)

# How to escape local minimum



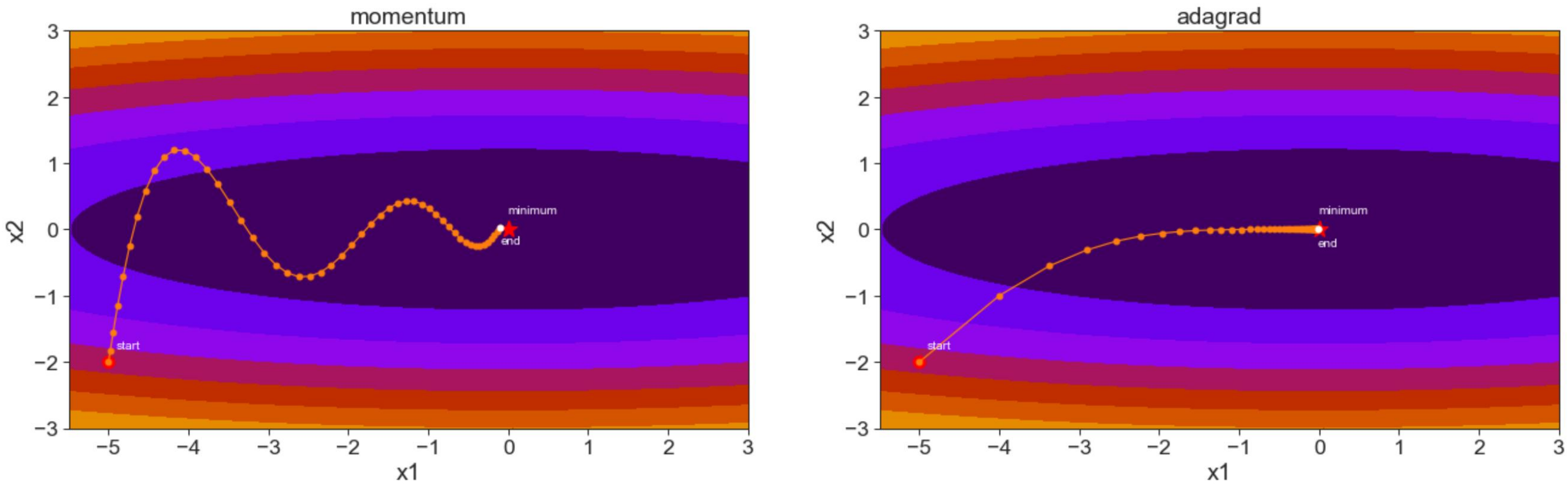

**1. Yann Lecun:** P(minimum) = $0.5^n$, more saddle points
2. SGD introduces disturbance
3. Momentum helps to escape local minimum or saddle point

# Momentum




(a) SGD (b) Momentum Mechanism

# Adaptively chaning learning rate




$$g_t = \nabla_\theta l, \quad G = \sum_t g_t^2,$$

$$\theta = \theta - \frac{\alpha}{\sqrt{G + \epsilon}} \cdot g_t.$$

- Apply smaller lr to the big $\partial_{\theta_i} l$ direction
- SGD + Momentum + Adaptive LR → Adam

# Auto Differentiation

- Forward Mode

Introduce dual numbers: $x \to x + \dot{x}\mathbf{d}$

where $\mathbf{d}^2 = 0$

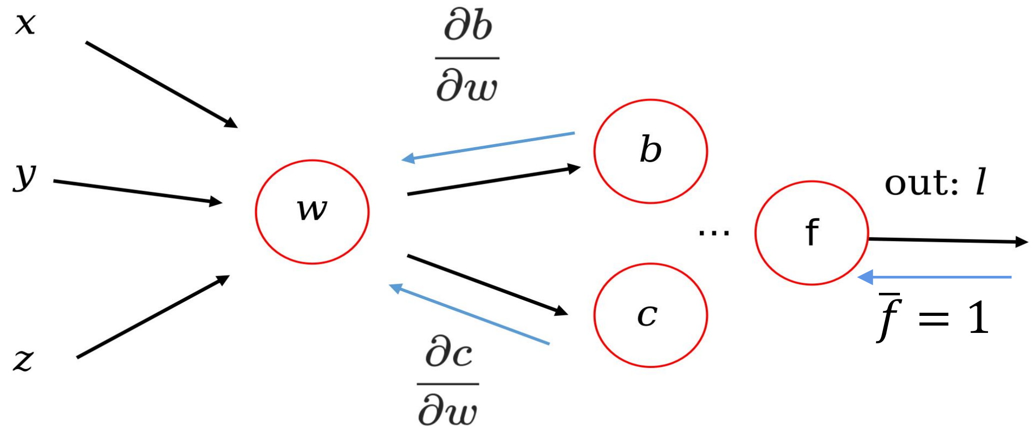
$$(x+\dot{x}\mathbf{d}) + (y+\dot{y}\mathbf{d}) = x + y + (\dot{x}+\dot{y})\mathbf{d}$$
$$(x+\dot{x}\mathbf{d}) - (y+\dot{y}\mathbf{d}) = x - y + (\dot{x}-\dot{y})\mathbf{d}$$
$$(x+\dot{x}\mathbf{d}) * (y+\dot{y}\mathbf{d}) = xy + (x\dot{y}+\dot{x}y)\mathbf{d}$$
$$\frac{1}{x+\dot{x}\mathbf{d}} = \frac{1}{x} - \frac{\dot{x}}{x^2}\mathbf{d} \qquad (x \neq 0)$$

**Forward mode for** $R^1 \to R^n$

**Reverse mode for** $R^n \to R^1$

- Reverse Mode

adjoint number: $\bar{w} = \dfrac{\partial l}{\partial w}$




$$\text{step } 1: \bar{w} = 0$$
$$\text{step } 2: \bar{w} = \bar{w} + \bar{b}\frac{\partial b}{\partial w}$$
$$\text{step } 3: \bar{w} = \bar{w} + \bar{c}\frac{\partial c}{\partial w}$$

# Tools for AutoDiff

PyTorch

MindSpore

JAX

TensorFlow

```
import torch

a = torch.tensor([2., 3.], requires_grad=True)
b = torch.tensor([6., 4.], requires_grad=True)

Q = 3*a**3 - b**2

dQdQ = torch.tensor([1., 1.])
Q.backward(gradient=dQdQ)

print(9*a**2 == a.grad)
print(-2*b == b.grad)
```

Out:

```
tensor([True, True])
tensor([True, True])
```

# The importance of initialization

… training deep models is a sufficiently difficult task that most algorithms are strongly affected by the choice of initialization. The initial point can determine whether the algorithm converges at all, with some initial points being so unstable that the algorithm encounters numerical difficulties and fails altogether.

Deep Learning, 2016

# Initialization methods

For Tanh activation function, the Xavier (Glorot) initialization is widely used.

For Sigmoid activation, one may use normalized Xavier initialization,

$$\text{weights} \sim N[0, \sigma]$$

where $\sigma = \sqrt{6/(n+m)}$, where n is the number of input neurons and m is the number of output neurons of that layer.

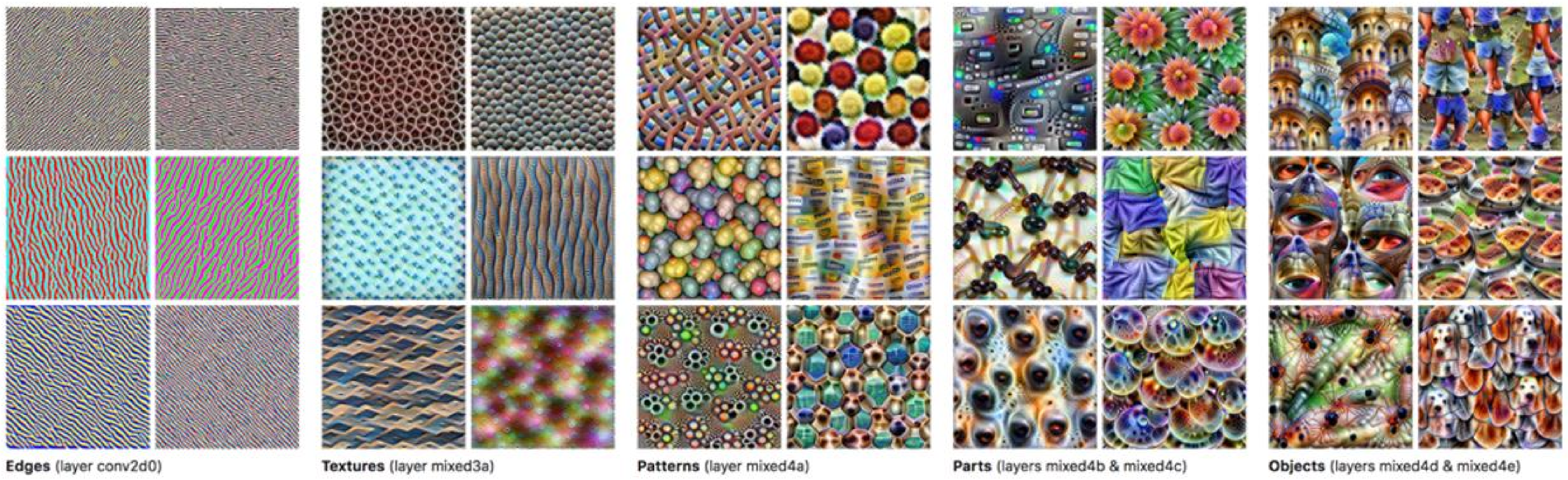
For ReLU activation function, one should use He initialization, e.g., in tensorflow,

$$\text{weights} \sim \text{U}\left[-\sqrt{\frac{6}{n}}, \sqrt{\frac{6}{m}}\right],$$

where U stands for uniform distribution,

# What has been learned （Global interpretation）

Olah, et al., "Feature Visualization", Distill, 2017.




shallow layers deep layers

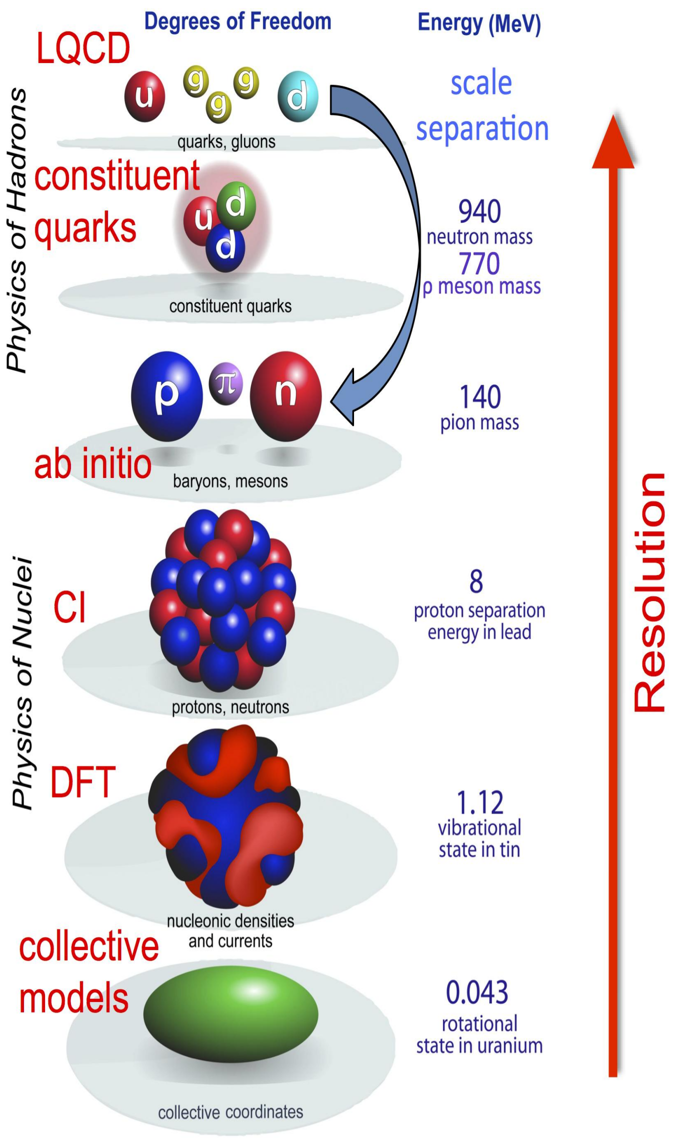
# Local interpretation

- **Ablation studies**： LIME or Prediction Difference Analysis. M. Tulio Ribeiro, et. al. "Why should I trust you?"

- **Class activation map**： map the deep layers to the input image， look for the most important region for decision making. BoLei Zhou, et. al. "Learning Deep Features for discriminative localization"

- **Layer-wise relevance propagation**： set the relavance of the output layer to 1， propagate the relevance to the input data, to look for the most important region for decision making.

# DL nuclear physics across energy scales




- **Deep generative models** (such as **normalizing flow** and the **diffusion model**) **have been used to sample Field Configureations in Lattice QCD**
- **Deep learning is widely used to solve inverse problems of HIC to study the EoS of hot QCD matter, the phase transition, the transport coefficients eta/s, …**
- **Deep neural network is used to represent the many-body wave function of nucleus, to solve variational problems in ab initio calculations**
- **Deep learning is used to solve inverse problems of HIC to study the nuclear structure, for instance, the nuclear deformation, neutron skin, alpha cluster and short range correlation**
- **DL for nuclear liquid droplet model…**

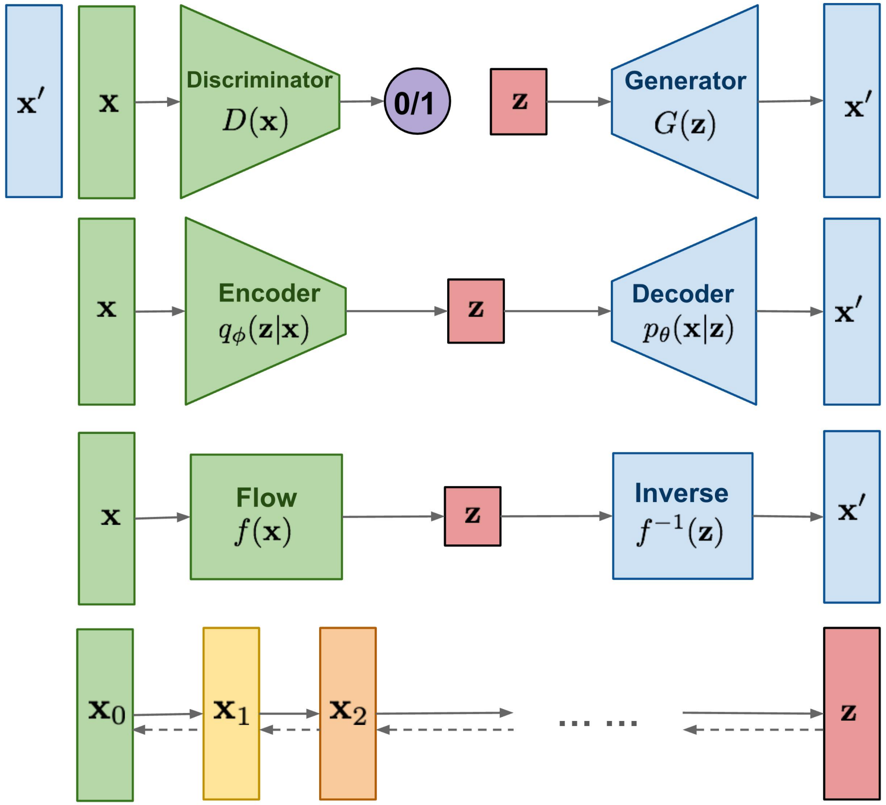
# Generativive models: MC sampling

GAN: Adversarial training

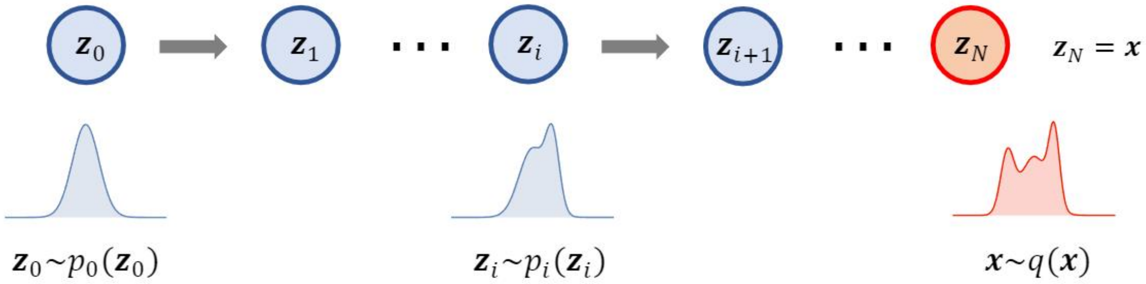
VAE: maximize variational lower bound

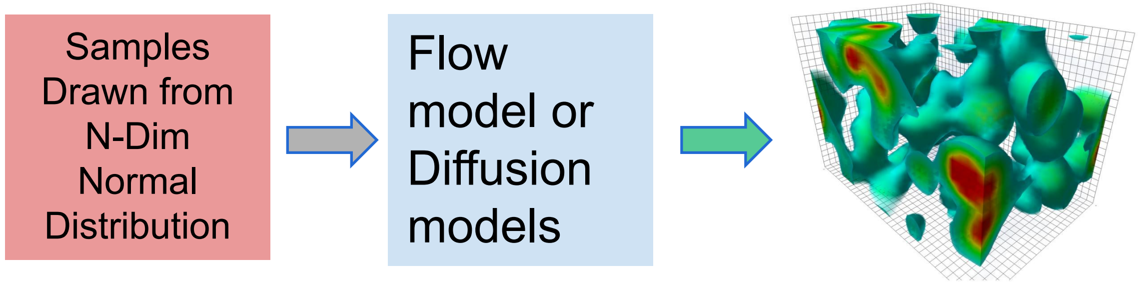
Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse




Similar to Box Muller algorithm




Flow-based generative models for Markov chain Monte Carlo in lattice field theory
Albergo, Kanwar, Shanahan 1904.1207

# Reviews

## *Colloquium*: Machine learning in nuclear physics

Amber Boehnlein, Markus Diefenthaler, Nobuo Sato, Malachi Schram, Veronique Ziegler, Cristiano Fanel
Morten Hjorth-Jensen, Tanja Horn, Michelle P. Kuchera, Dean Lee, Witold Nazarewicz, Peter Ostroumov,
Orginos, Alan Poon, Xin-Nian Wang, Alexander Scheinker, Michael S. Smith, and Long-Gang Pang

Rev. Mod. Phys. **94**, 031003 – Published 8 September 2022

Article | References | No Citing Articles | PDF | HTML | Export Citation

ABSTRACT

Advances in machine learning methods provide tools that have broad applicability in scientific resea
These techniques are being applied across the diversity of nuclear physics research topics, leading

## High energy nuclear physics meets Machine Learning

#1

Wan-Bing He (Fudan U., Shanghai and Fudan U.), Yu-Gang Ma (Fudan U., Shanghai and Fudan U.), Long-Gang Pang, Huichao Song (CCNU, Wuhan, Inst. Part. Phys. and Hua-Zhong Normal U., LQLP and Peking U.), Kai Zhou (Frankfurt U., FIAS) (Mar 12, 2023)

e-Print: 2303.06752 [hep-ph]

## Exploring QCD matter in extreme conditions with Machine Learning

Kai Zhou (Frankfurt U., FIAS), Lingxiao Wang (Frankfurt U., FIAS), Long-Gang Pang (CCNU, Wuhan, Inst. Part. Phys.), Shuzhe Shi (Stony Brook U.)

Mar 27, 2023

146 pages

e-Print: 2303.15136 [hep-ph]

## HEPML-LivingReview

### A Living Review of Machine Learning for Particle Physics

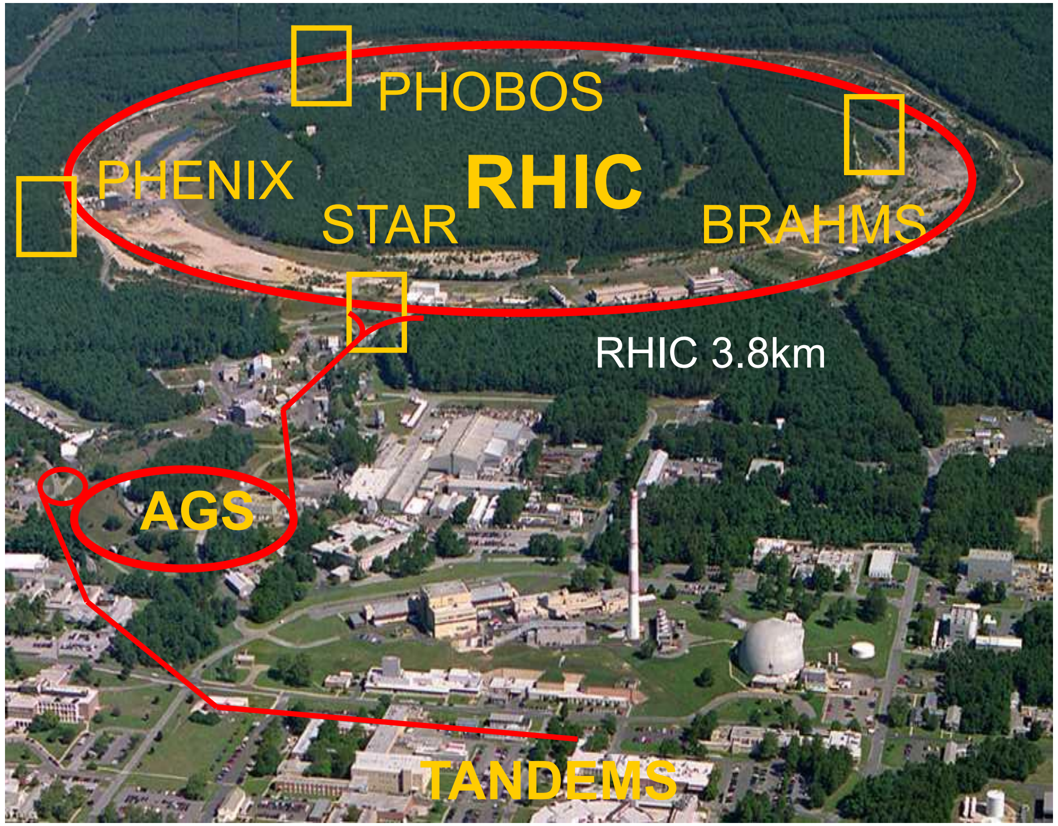
*Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.*

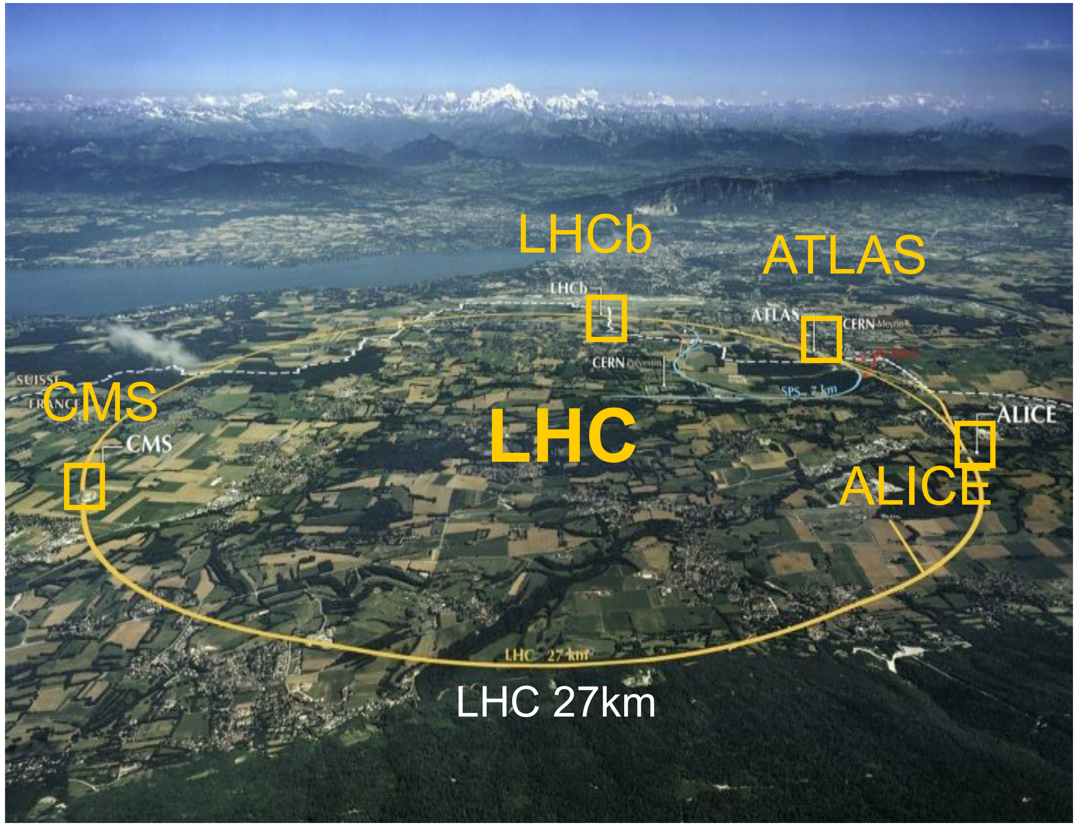
download review

The purpose of this note is to collect references for modern machine learning as applied to particle physics. A minimal number of categories is chosen in order to be as useful as possible. Note that papers may be referenced in more than one category. The fact that a paper is listed in this document does not endorse or validate its content - that is for the community (and for peer-review) to decide. Furthermore, the classification here is a best attempt and may have flaws - please let us know if (a) we have missed a paper you think should be included, (b) a paper has been misclassified, or (c) a citation for a paper is not correct or if the journal information is now available. In order to be as useful as possible, this document will continue to evolve so please check back before you write your next paper. If you find this review helpful, please consider citing it using \cite{hepmllivingreview} in HEPML.bib.

- Reviews
  - Modern reviews
    - Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning [DOI]
    - Deep Learning and its Application to LHC Physics [DOI]

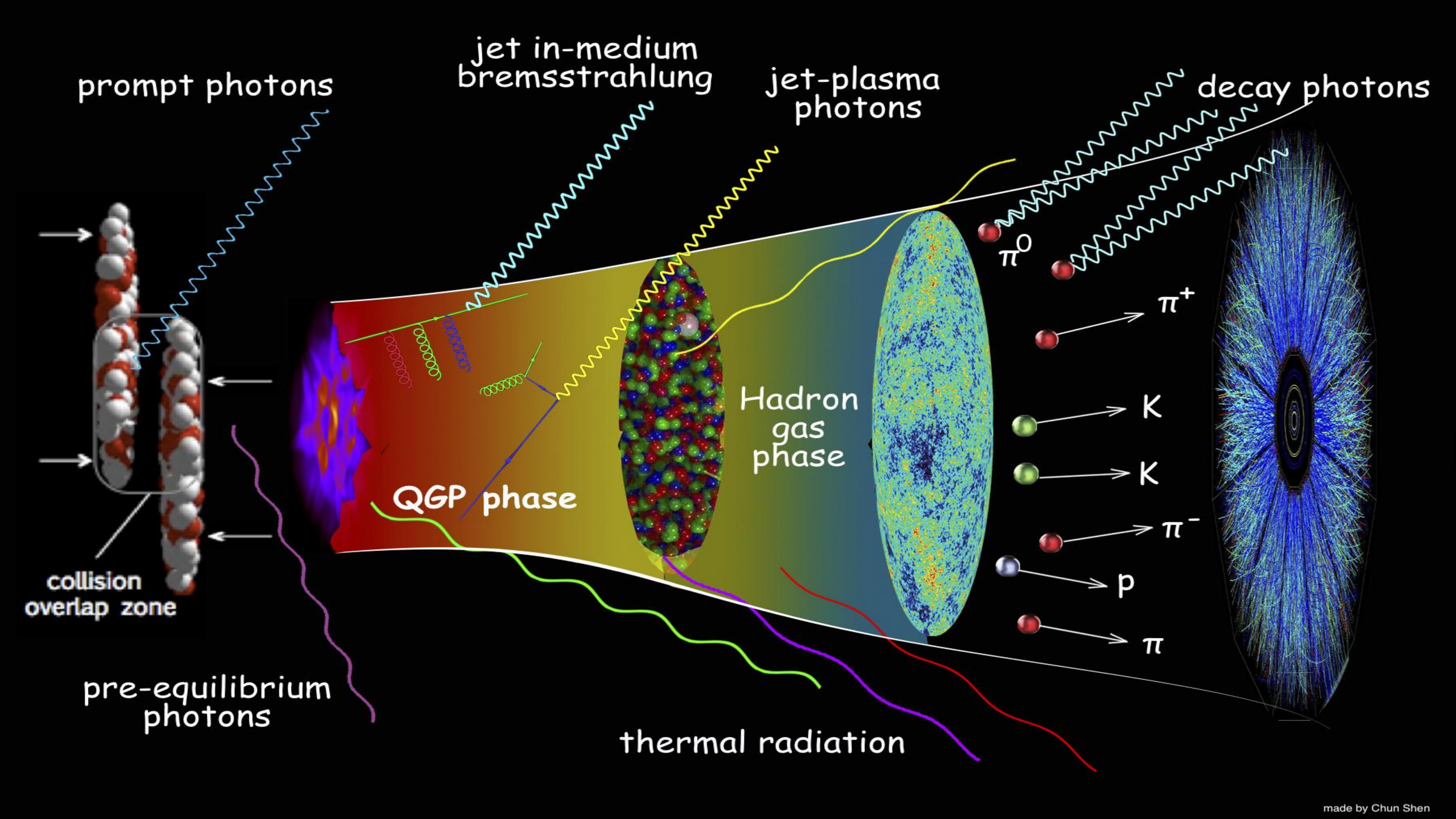
# New state of nuclear matter on earth



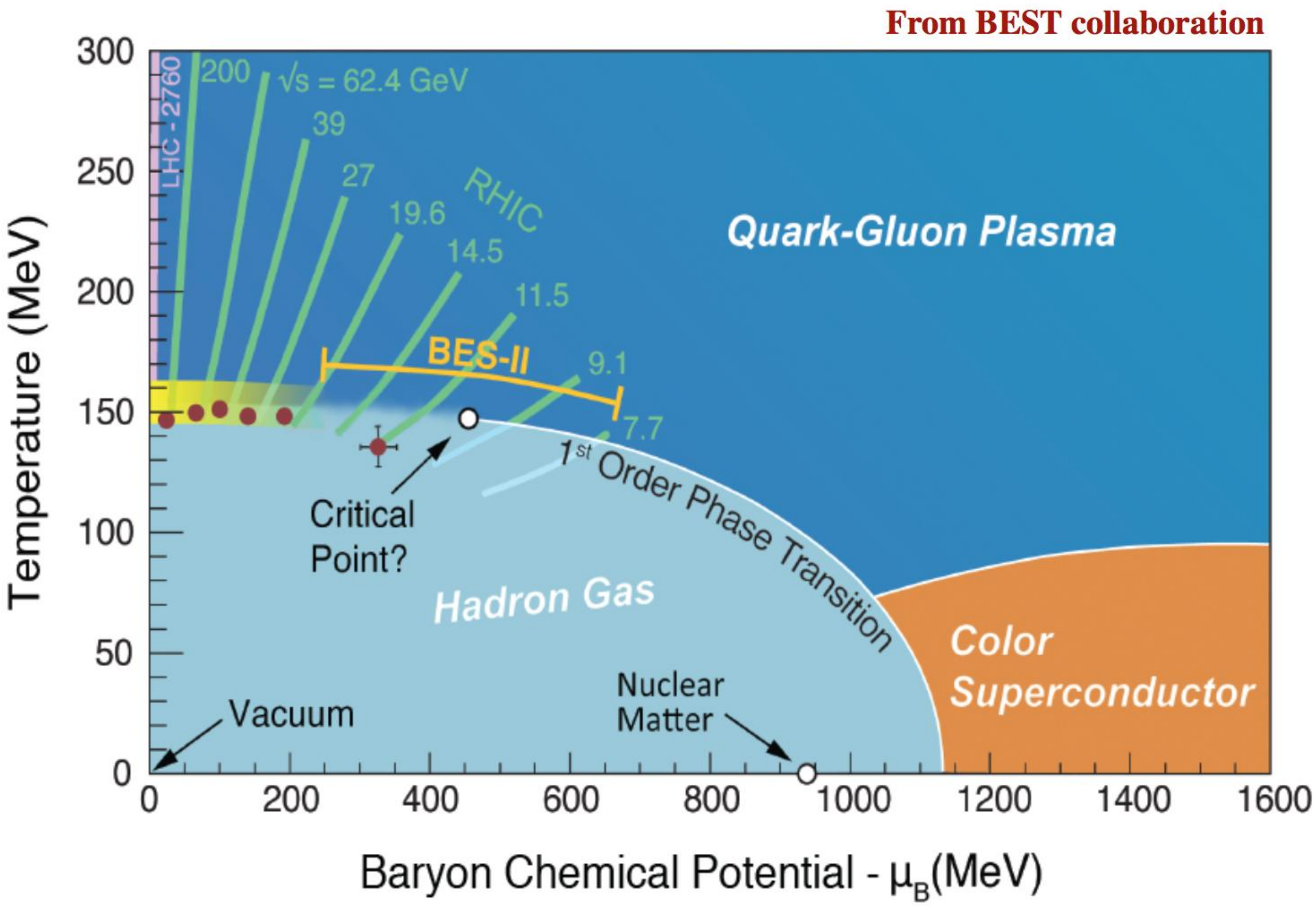




Au speed ～**99.99% c**

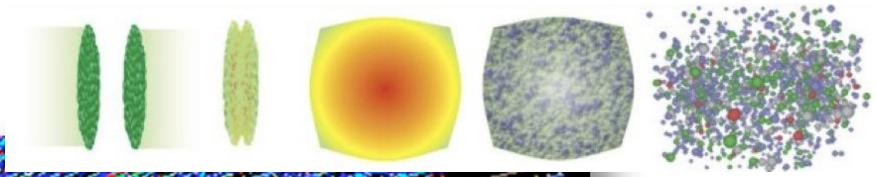
Pb speed～**99.9999% c**

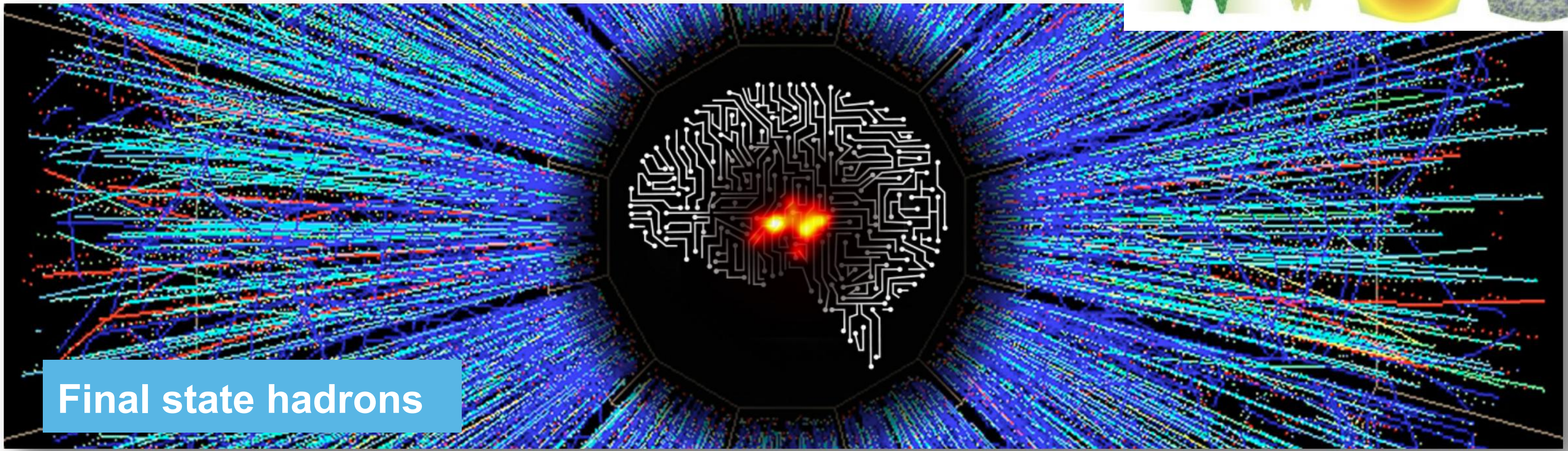

jet in-medium bremsstrahlung
prompt photons
jet-plasma photons
decay photons
$\pi^0$
$\pi^+$
K
K
$\pi^-$
p
$\pi$
Hadron gas phase
QGP phase
collision overlap zone
pre-equilibrium photons
thermal radiation
made by Chun Shen

# Explore QCD phase structure using HIC

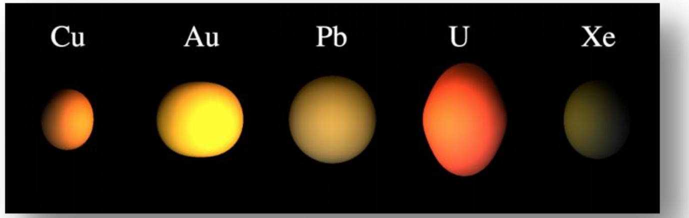


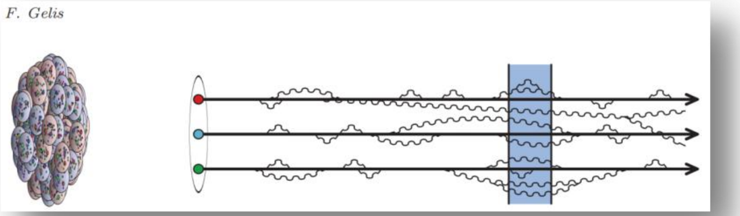
# Inverse problems in HIC

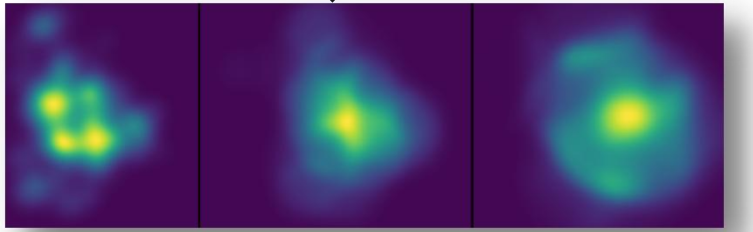





Non-linear mapping



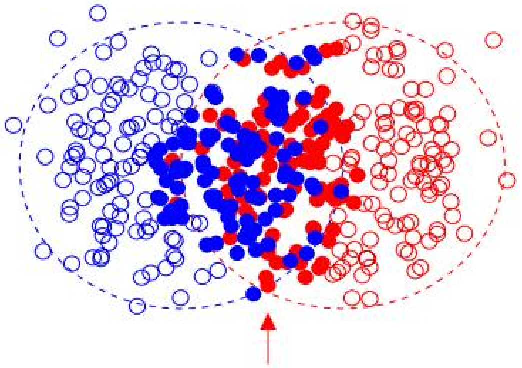



(1) Nuclear Structure

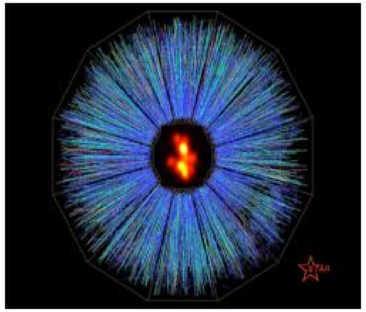
(2) Initial Parton Distribution

(3) QGP properties and EoS

# Theoretical model: relativistic hydro



$$\nabla_\mu T^{\mu\nu} = 0$$



$$T^{\mu\nu} = (\varepsilon + P)u^\mu u^\nu - P g^{\mu\nu} + \pi^{\mu\nu}$$

Initial condition　　　EoS　　　Viscosity

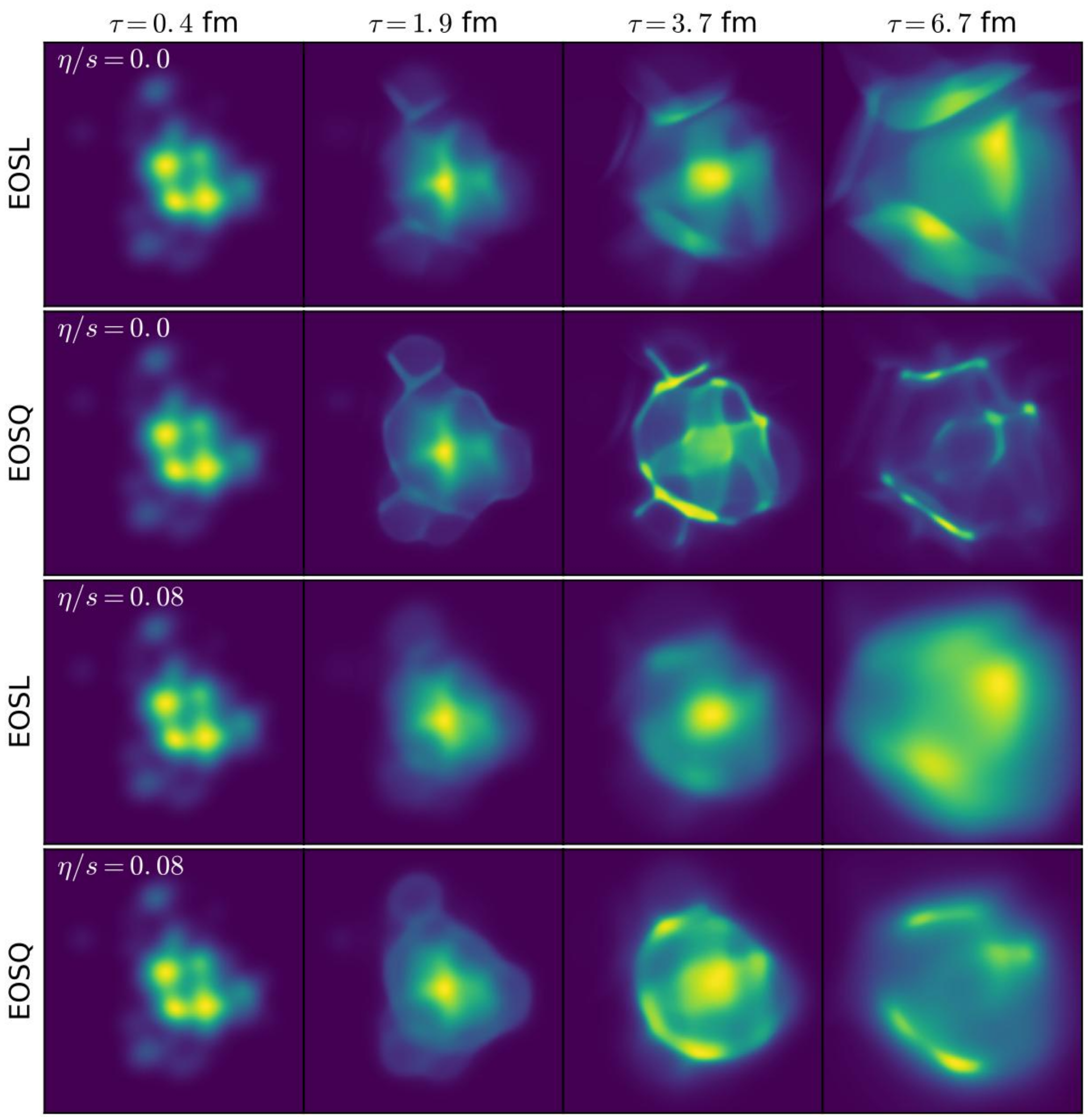
**Name of CLVisc**:

1. CCNU-LBNL Viscous Hydro, CCNU = Central China Normal University
2. A 3+1D viscous hydro parallized on GPU using OpenCL

**Purpose:** Describe the non-equilibrium space-time evolution of hot QCD matter
**Feature: 100 times faster** than using a single core CPU.

L.G. Pang, Q. Wang and X. N. Wang, PRC 86 (2012) 024911
L.G. Pang, B.W. Xiao, Y. Hatta, X.N.Wang, PRD 2015
L.G. Pang，H.Petersen, XN Wang, PRC97(2018)no.6,064918

# CLVisc for different EoS




**eta/s = 0**
**Lattice QCD EoS**
**(smooth cross over)**

**eta/s = 0**
**First order phase transition**

**eta/s = 0.08**
**Lattice QCD EoS**

**eta/s = 0.08**
**First order phase transition**

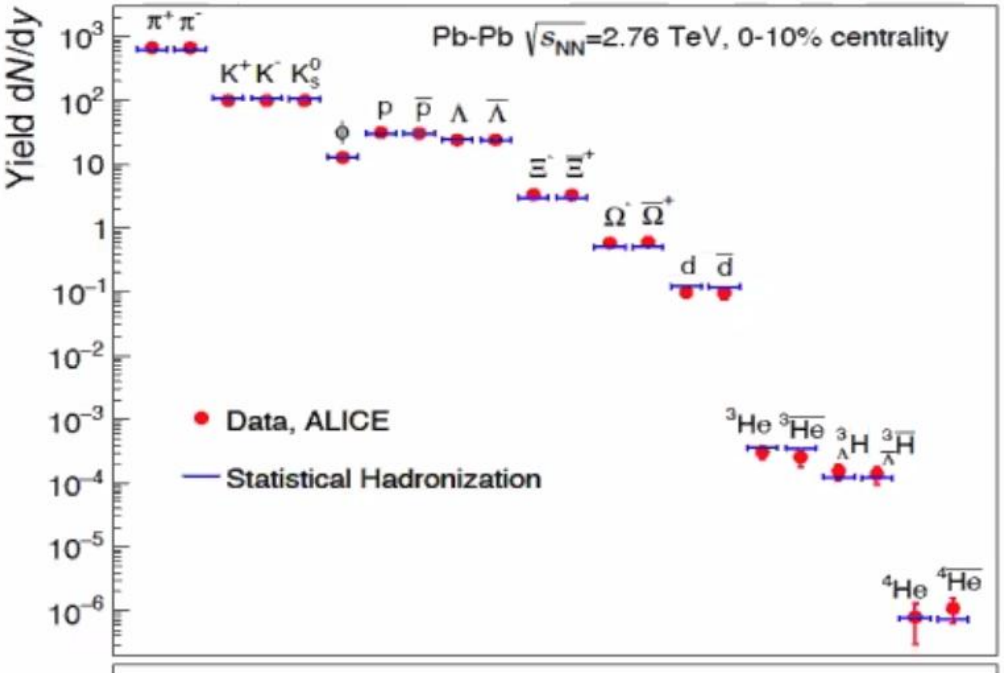
**eta/s: shear viscosity / entropy density**

Will the effect of EoS survive the dynamical evolution and exist in the final state hadrons?

# How many hadrons are produced

**statistical hadronization of (u,d,s) hadrons**

A. Andronic, P. Braun-Munzinger, K. Redlich, J. Stachel, Nature 561 (2018) 321




Best fit:

$T_{CF} = 156.6 \pm 1.7$ MeV

$\mu_B = 0.7 \pm 3.8$ MeV

$V_{\Delta y=1} = 4175 \pm 380$ fm$^3$

$\chi^2/N_{df} = 16.7/19$

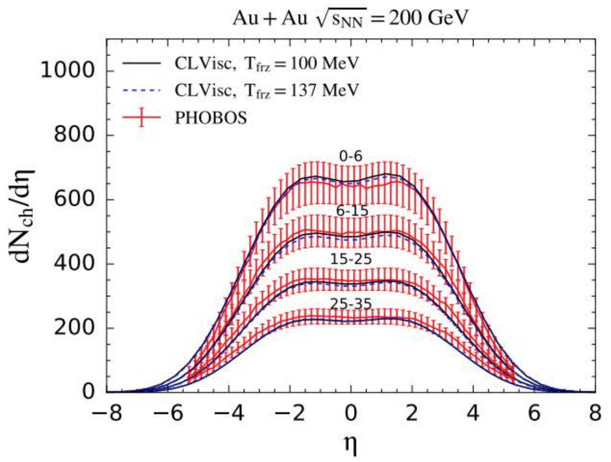
*S-matrix treatment of inter-*

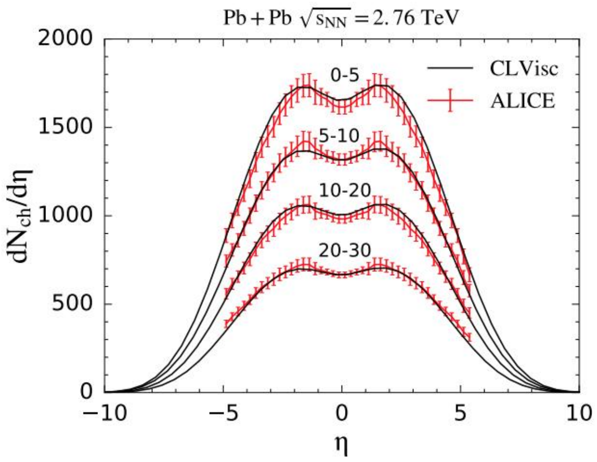
$$n_i = (2s_i + 1)4\pi \int p^2 \left[ e^{\left(\sqrt{p^2+m_i^2} - \mu_i\right)/T} \pm 1 \right]^{-1} dp$$

1. At LHC, equal amounts of matter and anti-matter are produced
2. At BES region, more protons than anti protons

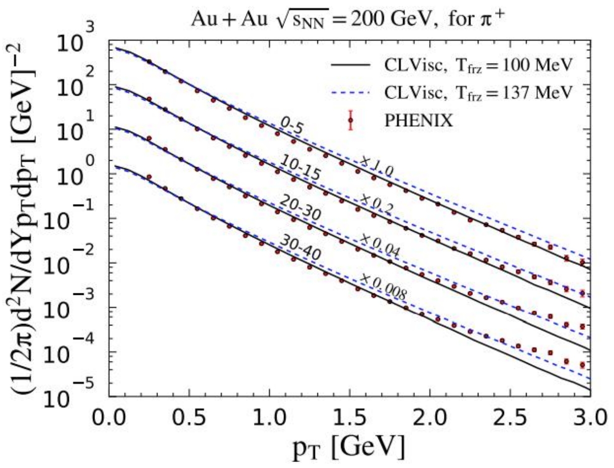
# CLVisc vs experimental data

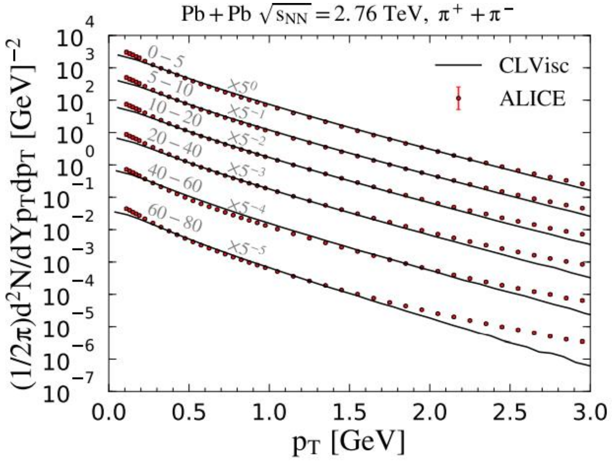
## Longitudinal momentum distribution



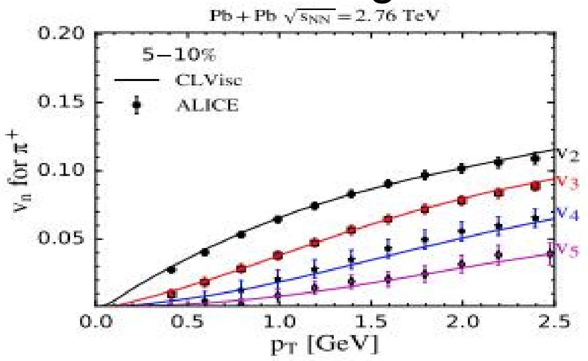



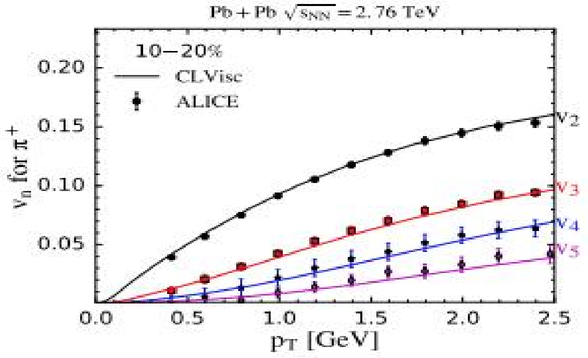

## Transverse momentum distribution

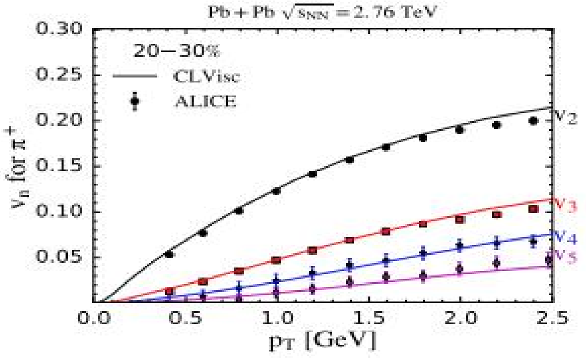



## Fourier decomposition coef. for azimuthal angle

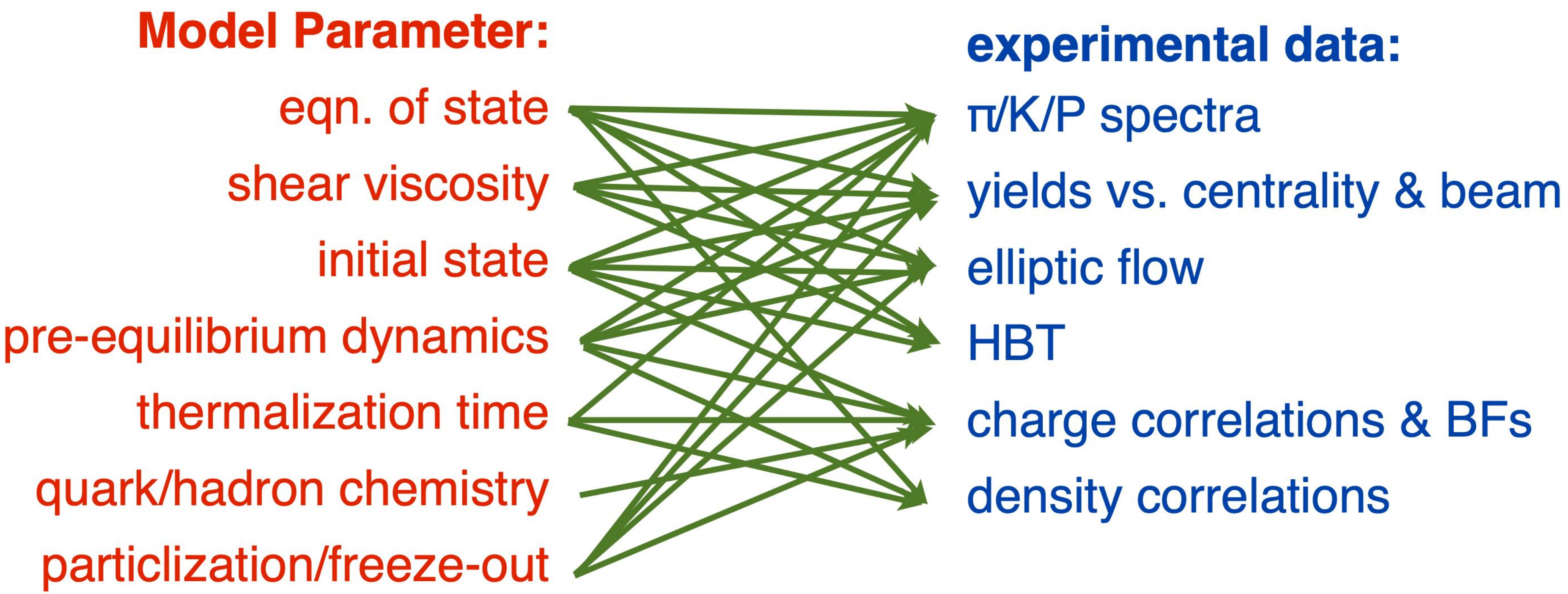


# Challenges

Fig from S. Bass QM2017 (Bayesian method)




**(1) Multiple parameters entangle with multiple observables**

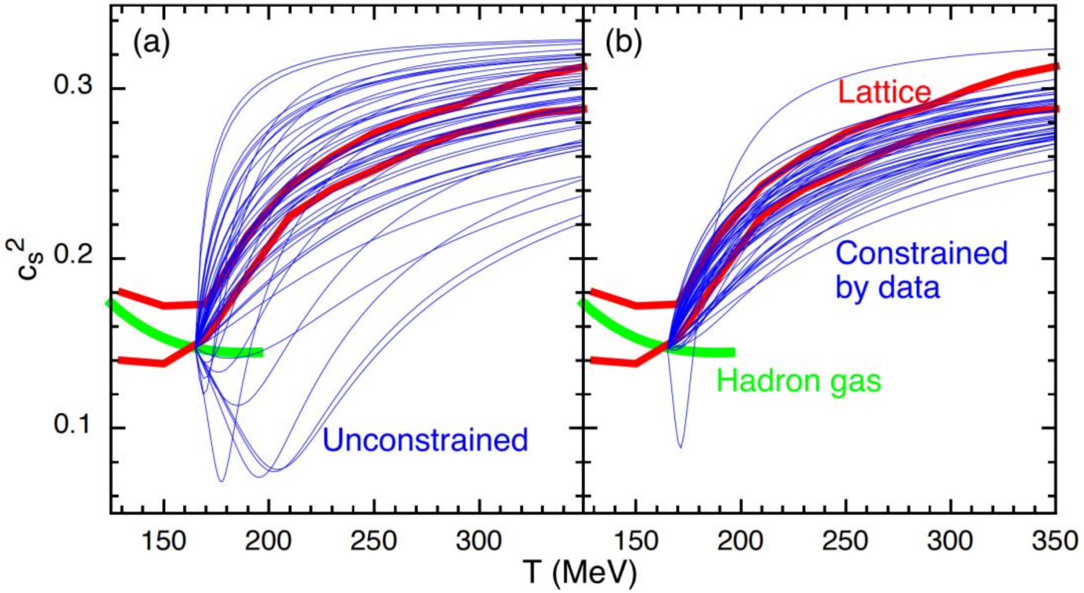
**(2) Different parameter combinations describe the same data**

# Bayesian analysis QCD EoS

The $c_s^2$ is parameterized as a function of energy density in the following,

$$c_s^2(\epsilon) = c_s^2(\epsilon_h) + \left(\frac{1}{3} - c_s^2(\epsilon_h)\right) \frac{X_0 x + x^2}{X_0 x + x^2 + X'^2} \tag{2.12}$$

where $X_0 = \sqrt{12} R X' c_s(\epsilon_h)$, $x \equiv \ln \frac{\epsilon}{\epsilon_h}$, $\epsilon_h$ is the energy density at $T = 165$ MeV, $R$ and $X'$ are the two parameters in the EoS to be determined. Randomly choosing $R$ and $X'$ from the range $-0.9 < R < 2$ and $0.5 < X' < 5$ generate the unconstrained EoS that varies in a large region between $c_s^2 = 0.05$ and $c_s^2 = 0.33$, as shown in Fig. 2.4-a. This corresponds to the a priori distribution of $c_s^2$ parameters together with other 12 parameters in the model $P(\theta)$.

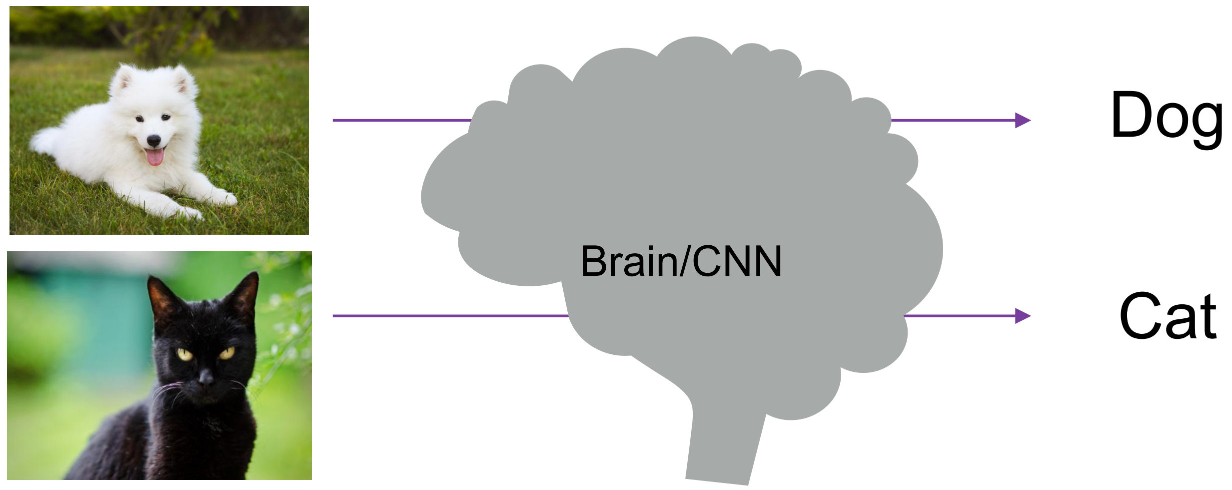


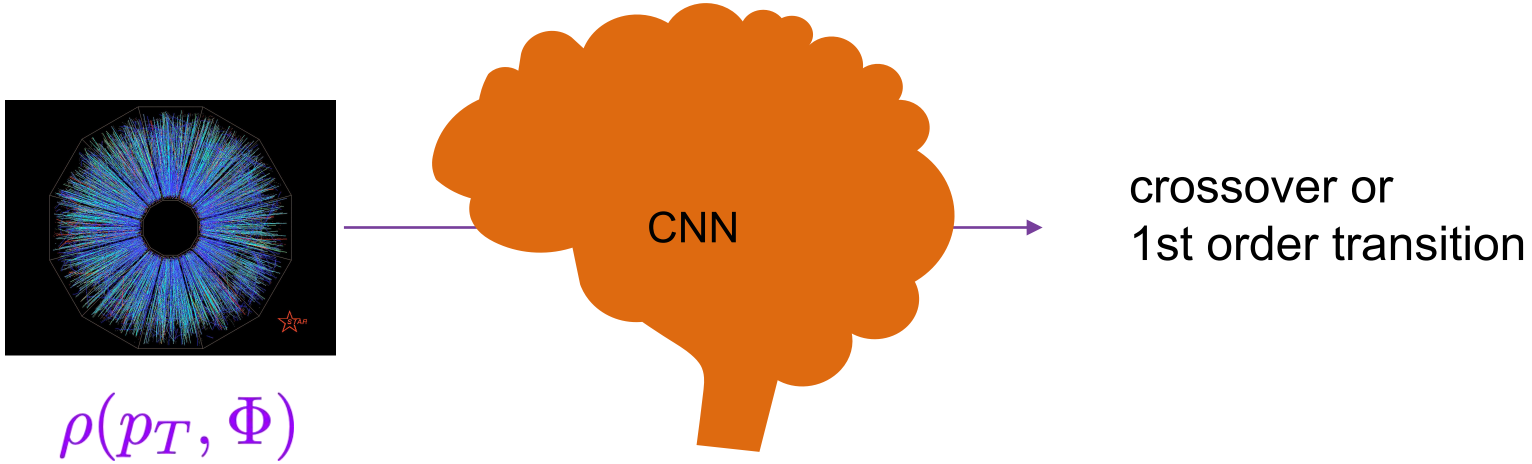

Likelihood: $$P(D|\theta) = \Pi_i \exp\left(-(z_i(\theta) - z_{i,\exp})^2/2\right)$$

Posterior: $$P(\theta \mid D) \propto P(D \mid \theta) P(\theta)$$

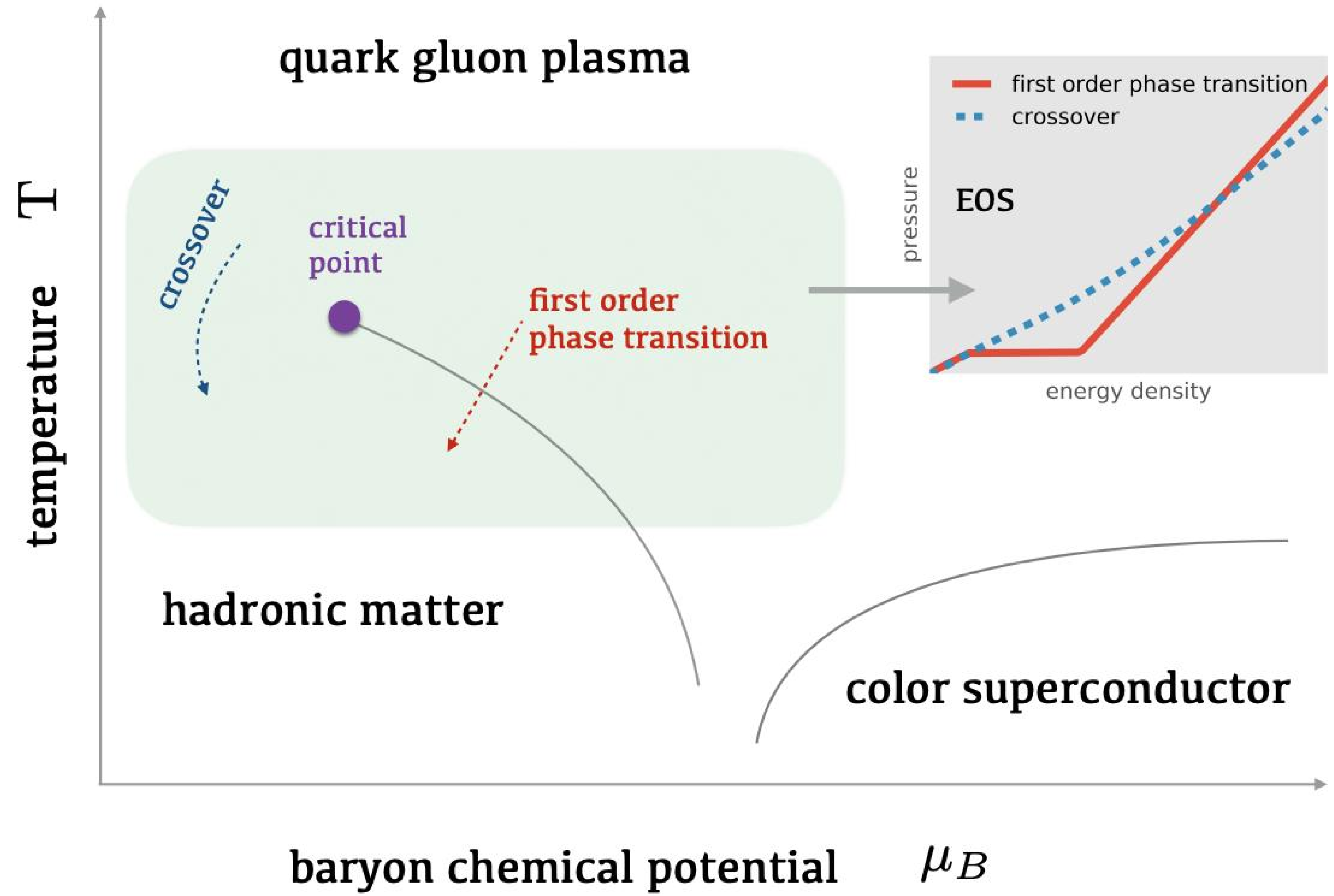
S. Pratt, E. Sangaline, P. Sorensen, H. Wang, PRL. 114 (2015) 202301.

# Excellent pattern recognition of deep learning

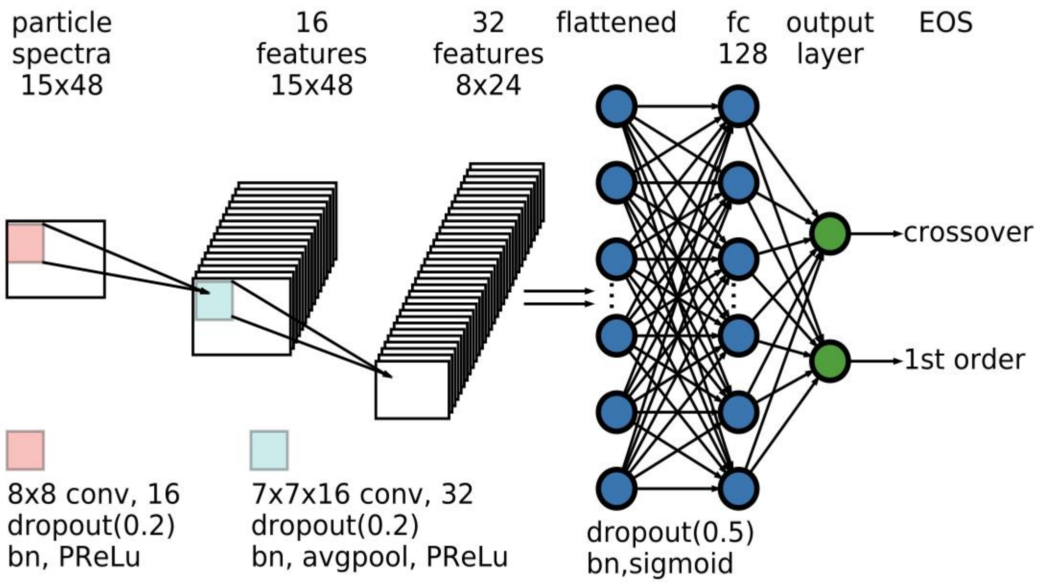


# EoS for different phase transition types

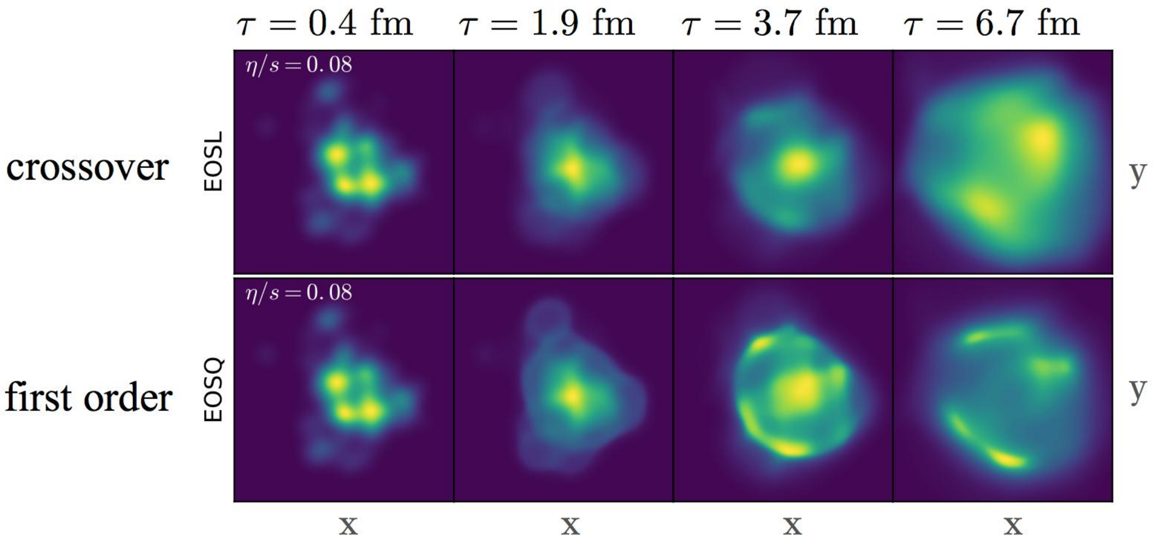


# Determine nuclear phase transitions




$$l(\theta) = \underbrace{-\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)]}_{\text{cross entropy loss}} + \underbrace{\lambda||\theta||_2^2}_{\text{L2 regularization}}$$

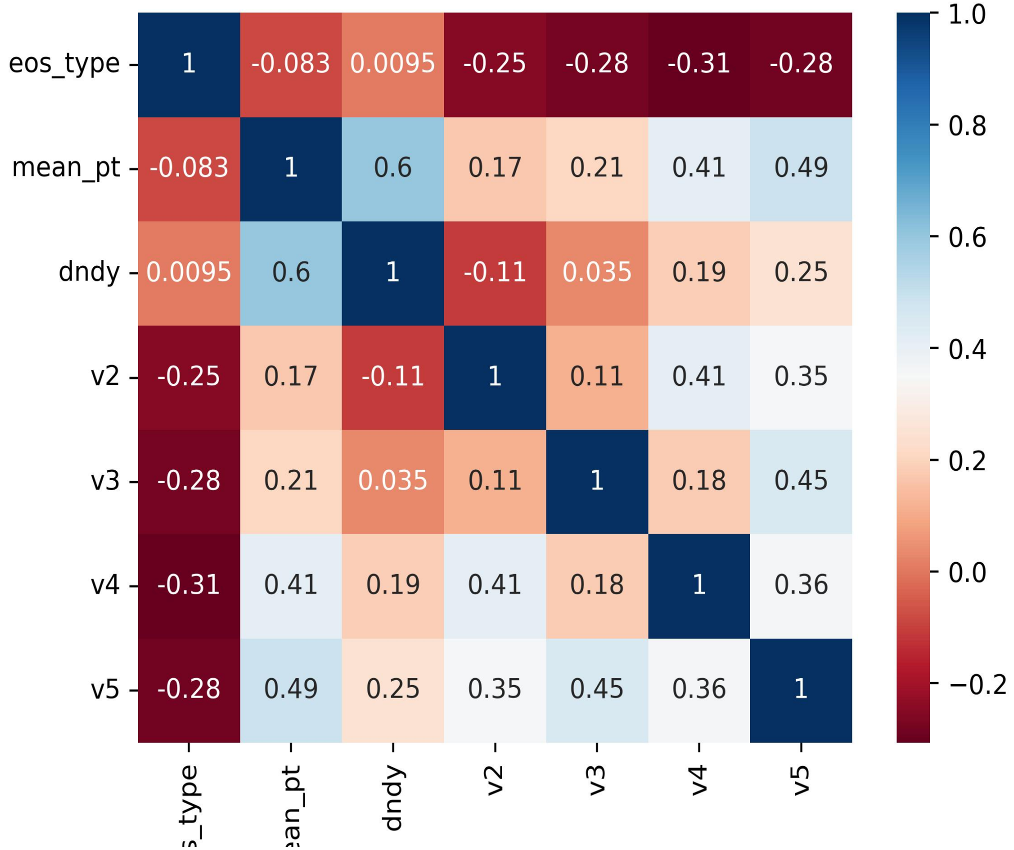
$$\nabla_\mu T^{\mu\nu} = 0$$

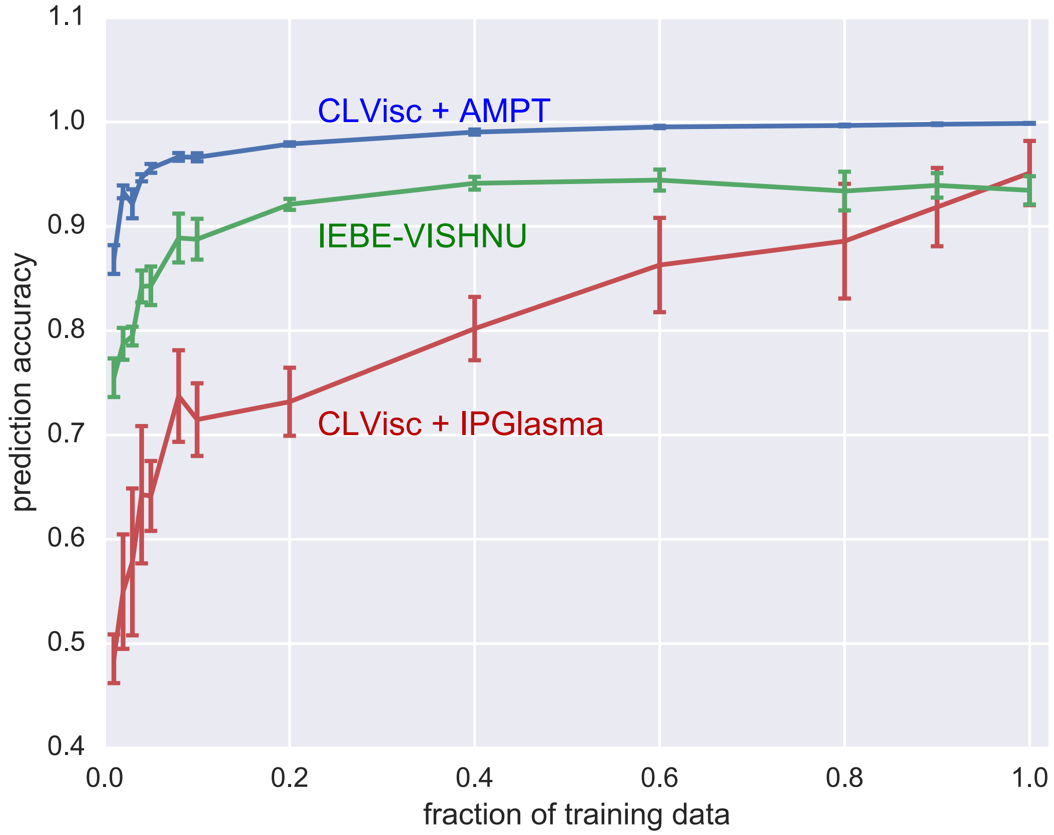


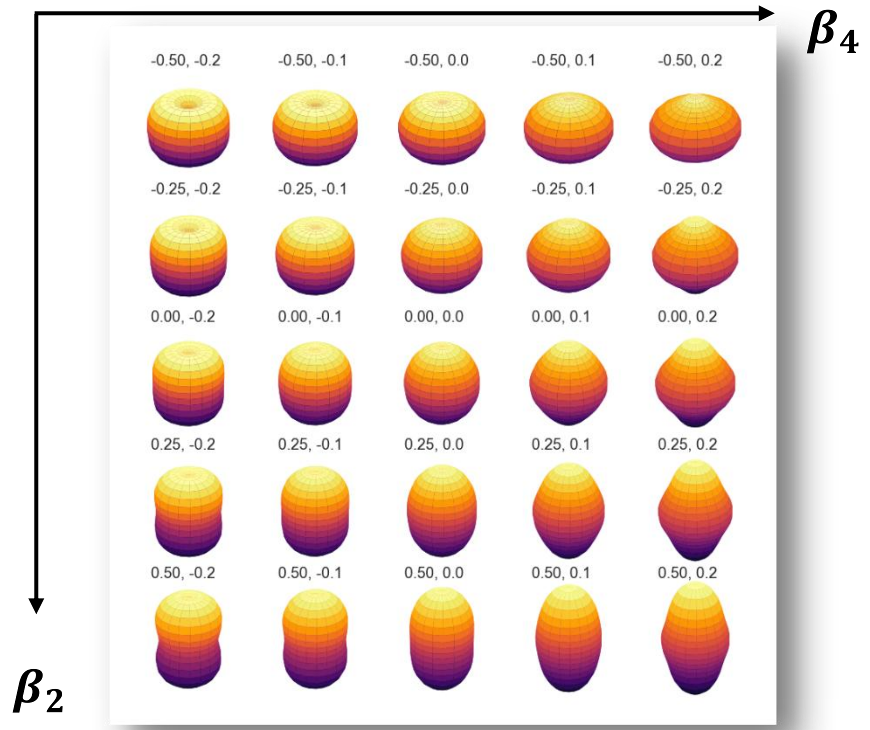

CLVisc 3+1D relativistic hydrodynamics

Nature Communications 2018, **LG. Pang**, K.Zhou, N.Su, H.Petersen, H. Stoecker, XN. Wang.

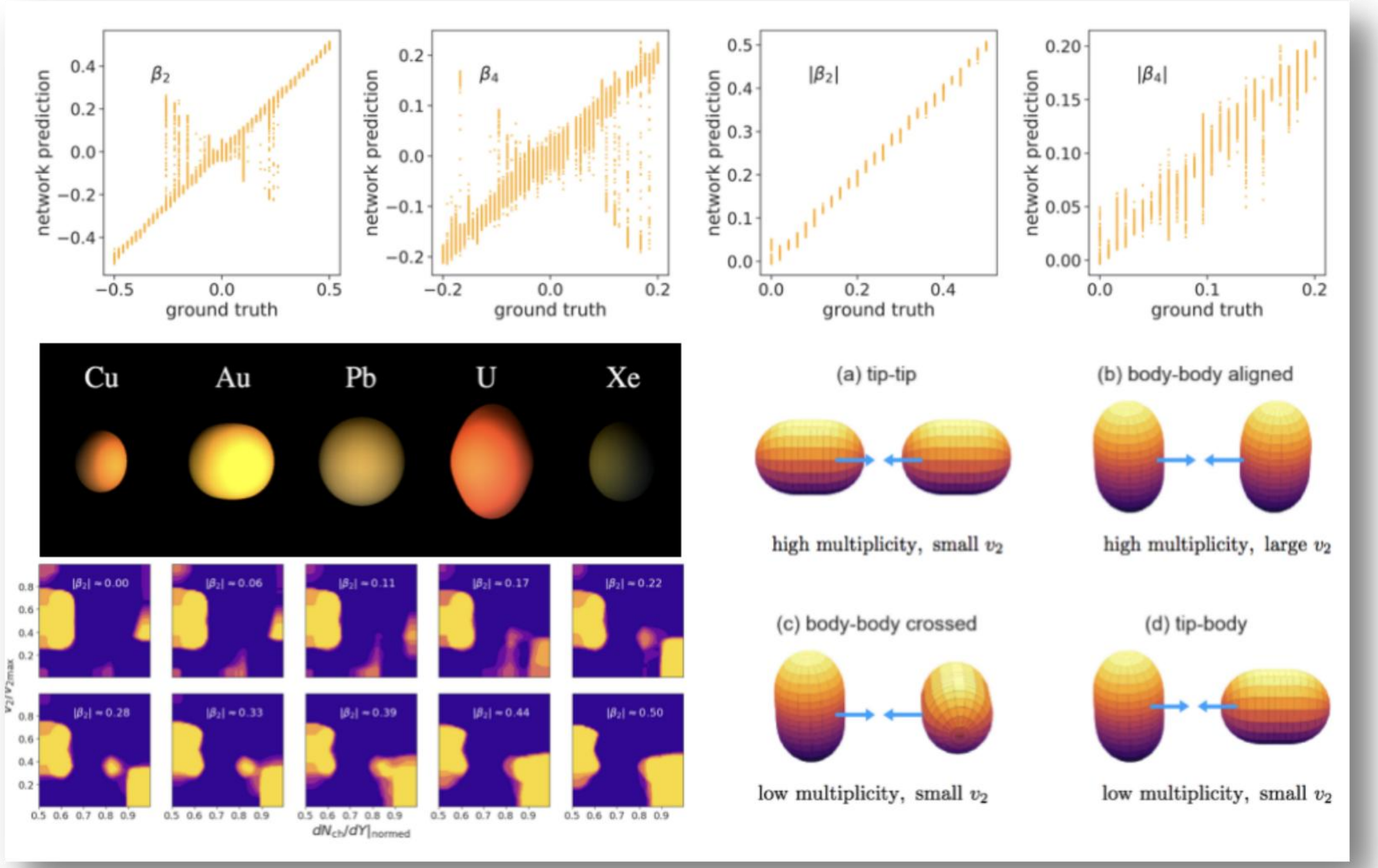
# DL with CNN for EoS classification



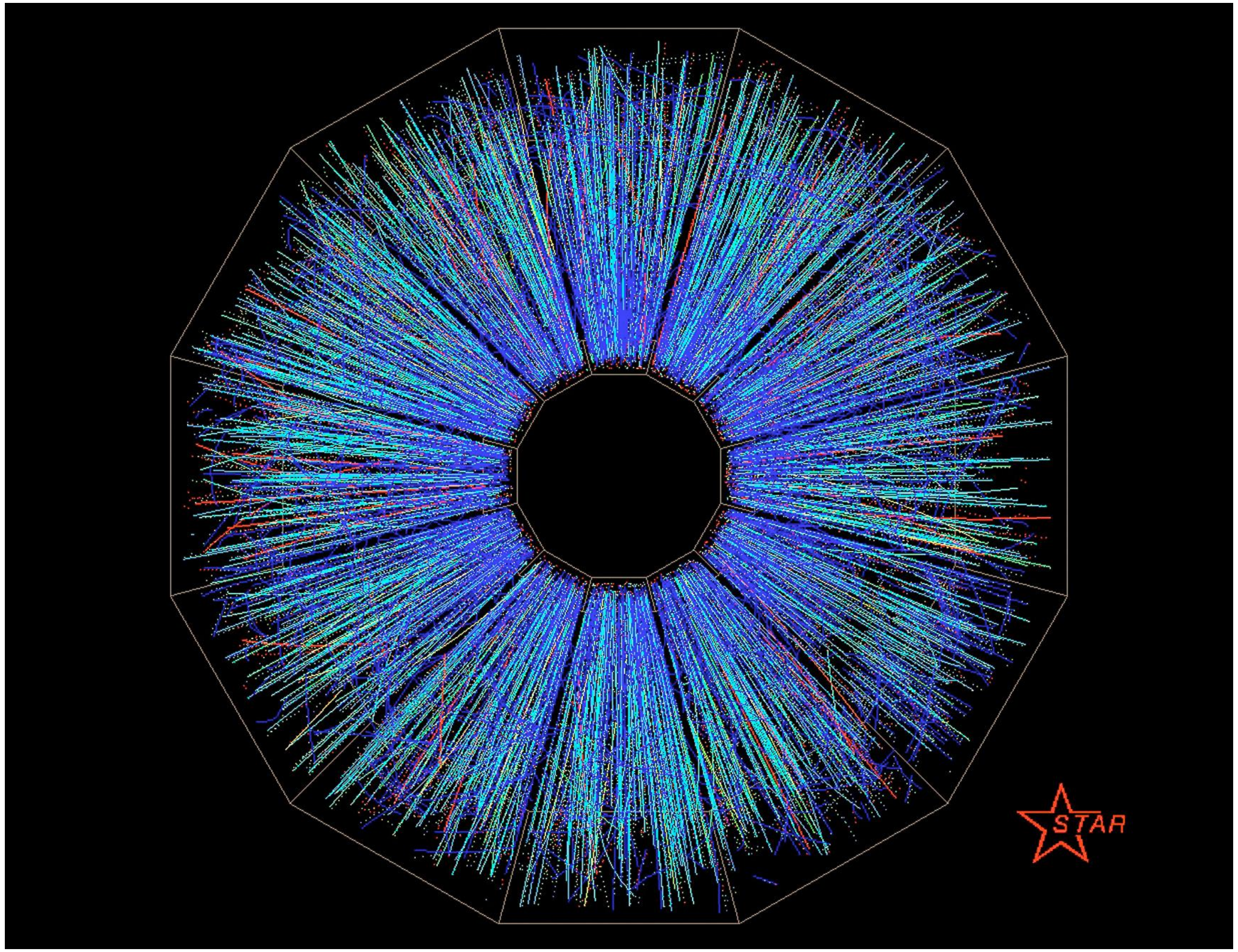



**Data: Trento + Matching**




L.-G. Pang, K. Zhou and X.-N. Wang, arXiv:1906.06429

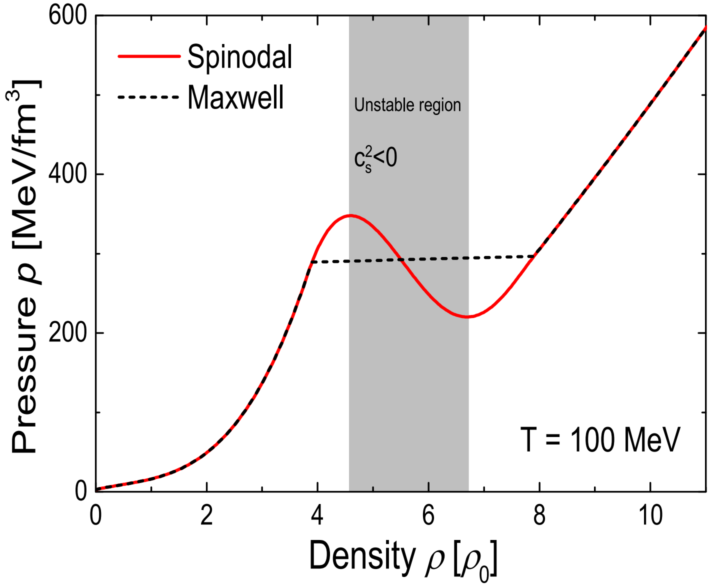
# Data representation

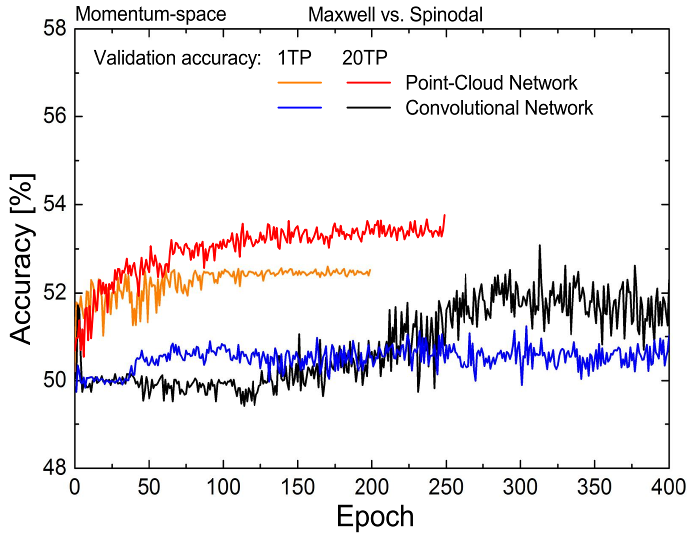


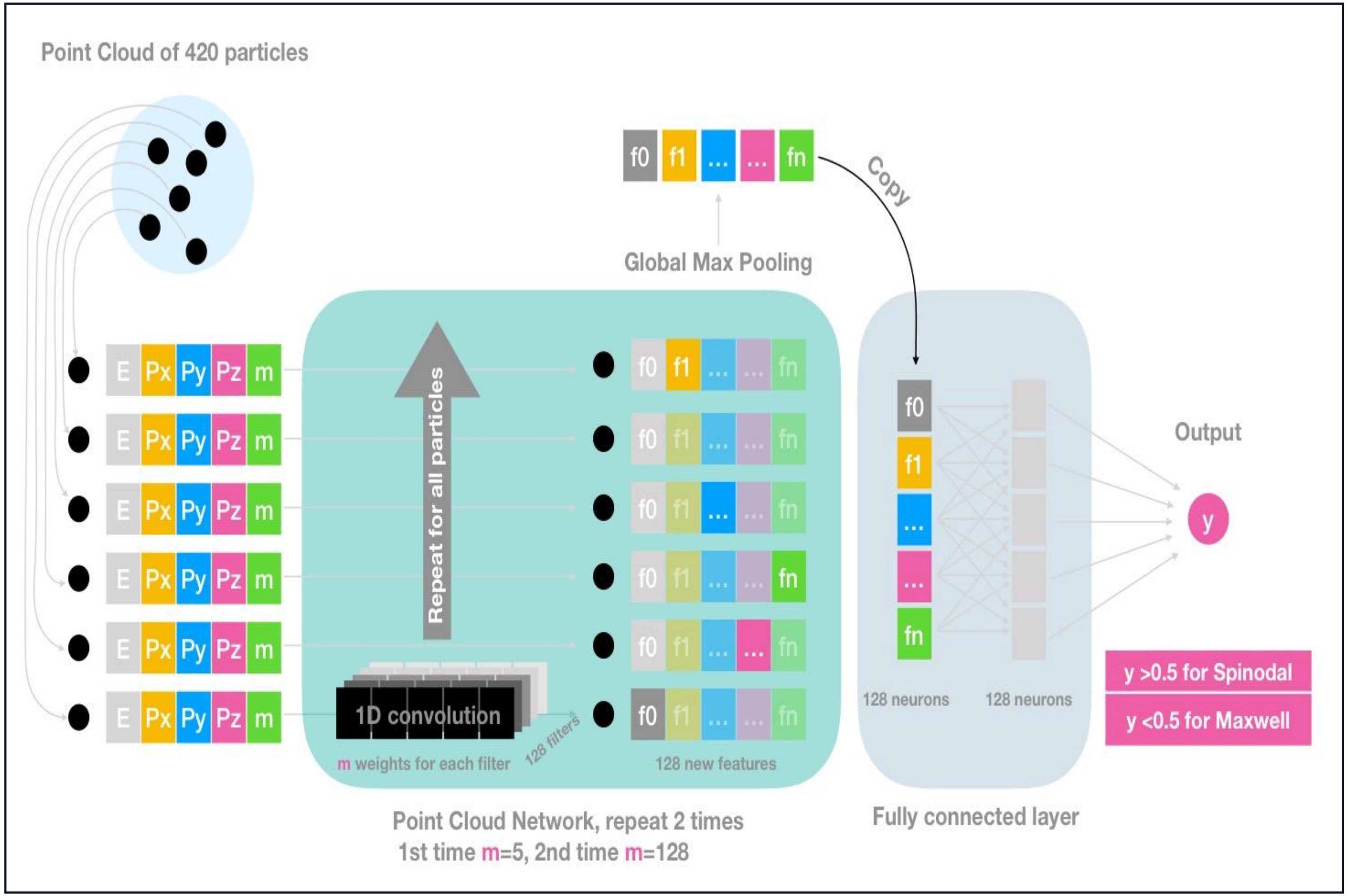

- Images: histograms
  - (px, py) or (pt, phi)
  - (px, py, pz)
  - (pt, phi, eta)
- Point cloud: particle list

| E | Px | Py | Pz | pid |
|---|---|---|---|---|
| 6.84 | 1.07 | 4.5 | 6.83 | 211 |
| 68.92 | 0.75 | 0.64 | 68.91 | 2212 |
| 40.4 | 0.06 | 0.54 | 40 | 321 |
| … | | | | |

# Spinodal vs Maxwell 1st order phase transition




J. Steinheimer, L.G. Pang, K. Zhou, V. Koch and J. Randrup, JHEP 12 (2019) 122

## Dynamical Edge Convolution Network

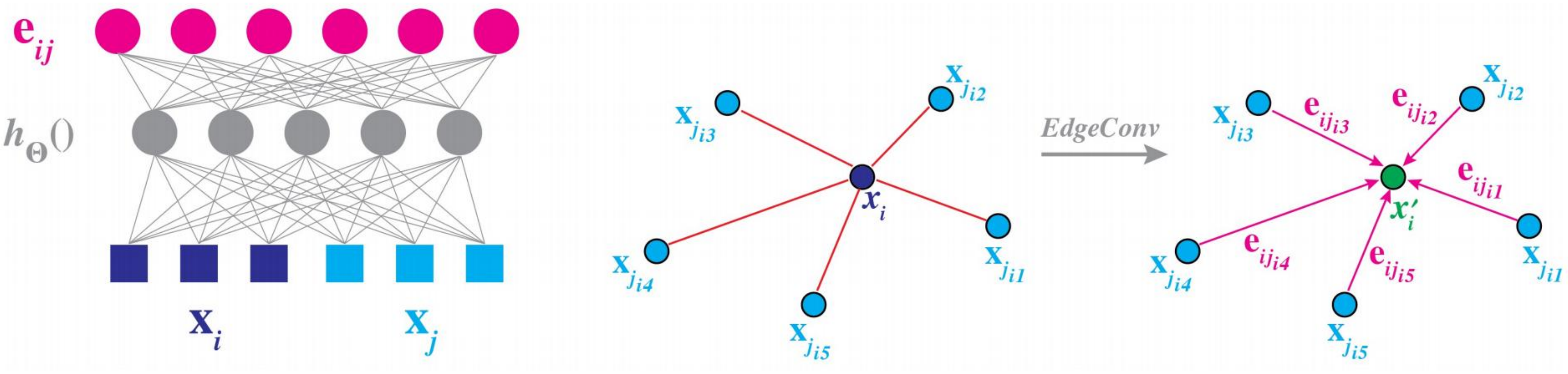
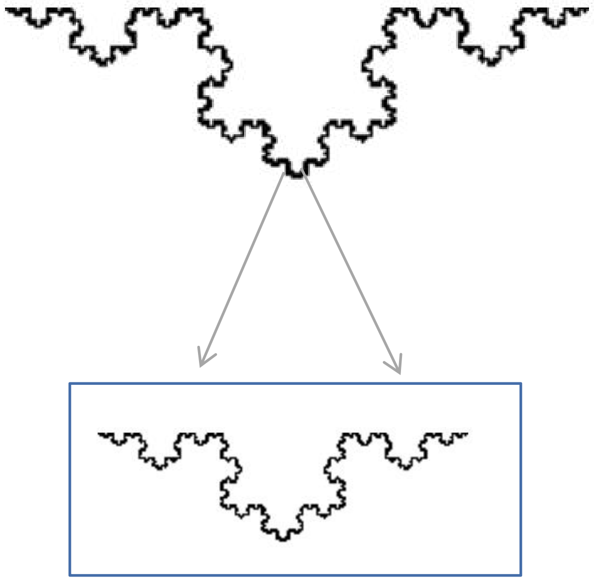


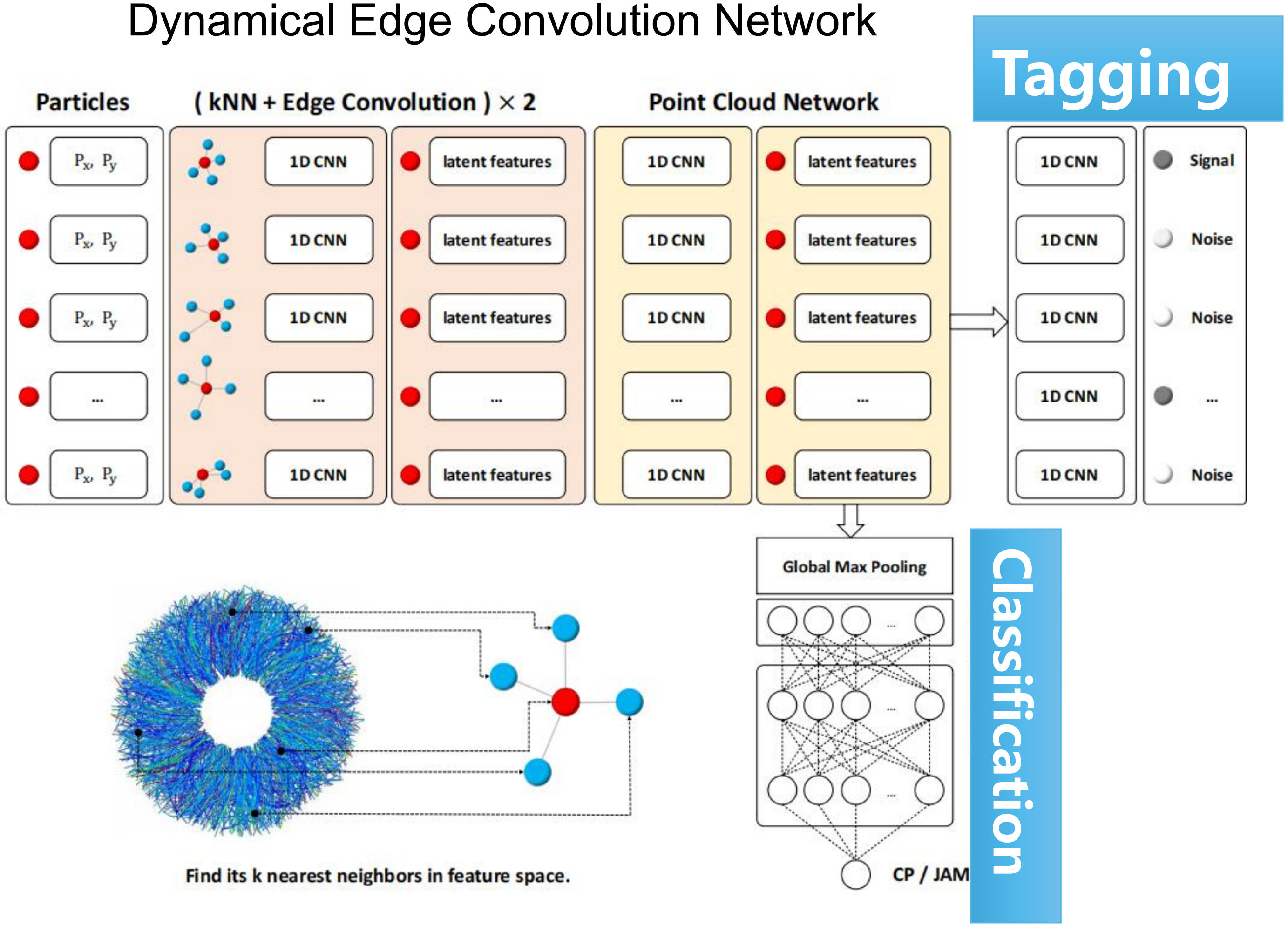

Fig. 2. **Left**: Computing an edge feature, $\mathbf{e}_{ij}$ (top), from a point pair, $\mathbf{x}_i$ and $\mathbf{x}_j$ (bottom). In this example, $h_\Theta()$ is instantiated using a fully connected layer, and the learnable parameters are its associated weights. **Right**: The EdgeConv operation. The output of EdgeConv is calculated by aggregating the edge features associated with all the edges emanating from each connected vertex.

# Looking for self similarity in momentum space



Self similarity, scaling invariance

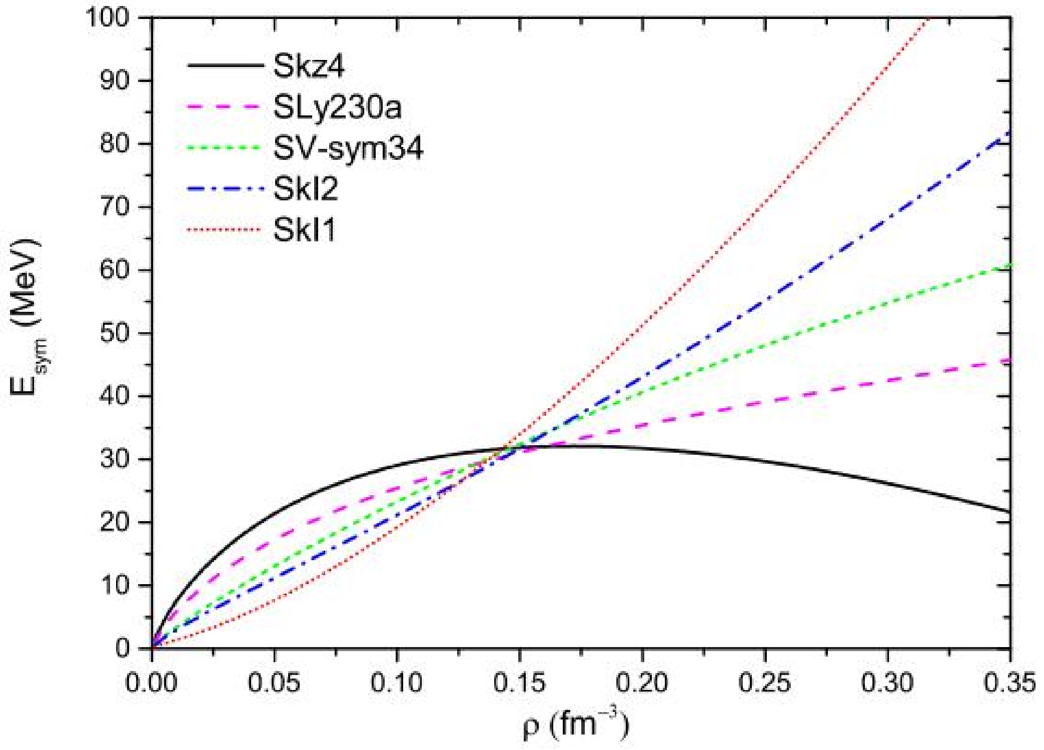
Dynamical Edge Convolution Network



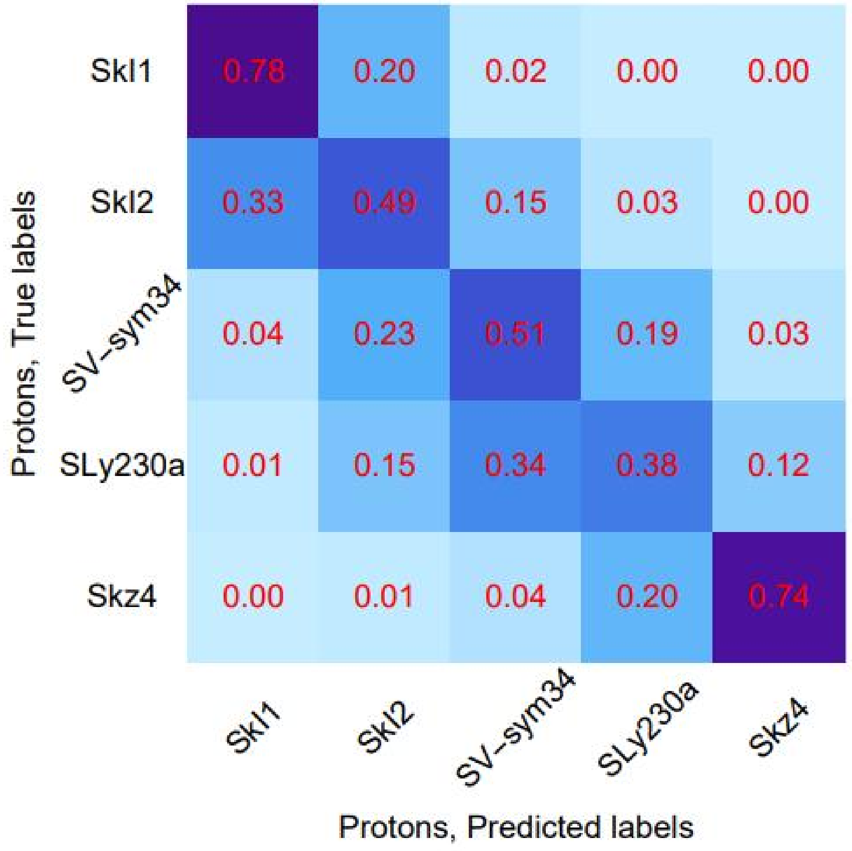

PLB 827(2022) 137001, Y.-G. Huang, L.-G. Pang, X.F. Luo and X.-N. Wang

# Nuclear EoS at high density region

Skyrme potential + IMQMD




off-diagonal = misclassified



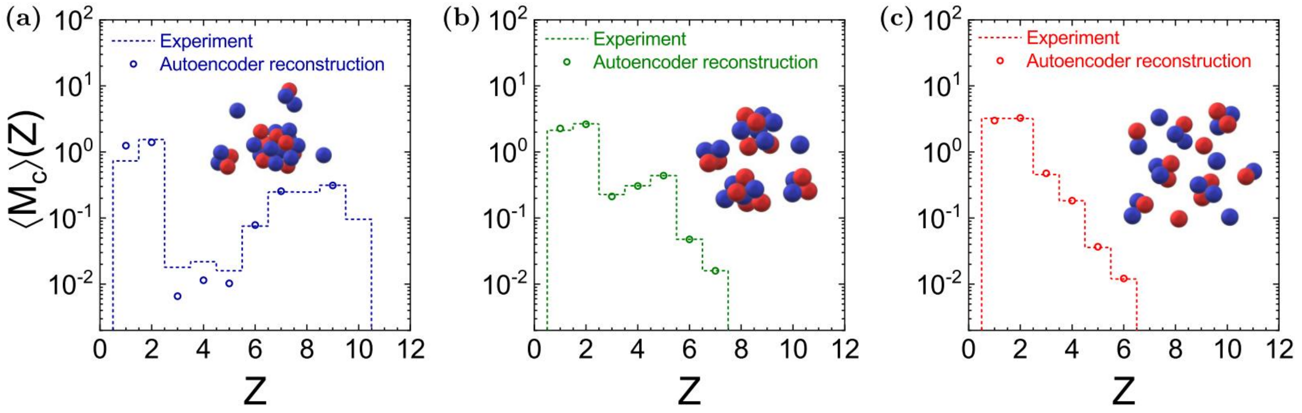

PLB 822 (2021) 136669, Y.J Wang, F.P. Li, Q.F. Li, H.L. L¨u, and K. Zhou

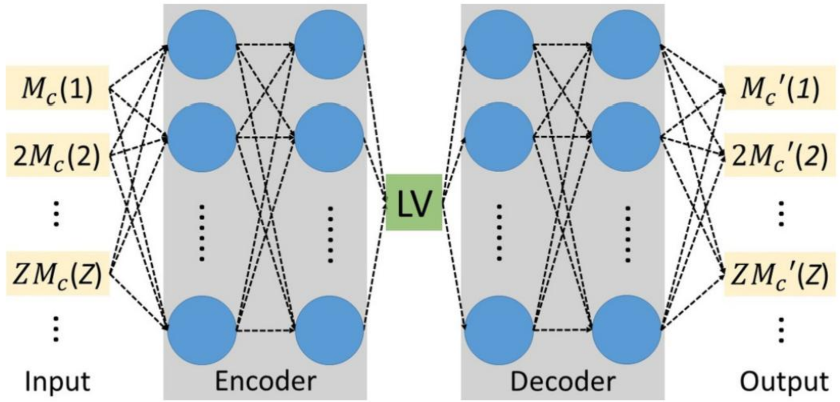
# Auto Encoder for order parameter

PHYSICAL REVIEW RESEARCH **2**, 043202 (2020)

**Nuclear liquid-gas phase transition with machine learning**

Rui Wang,[1,2,*] Yu-Gang Ma,[1,2,†] R. Wada,[3] Lie-Wen Chen,[4] Wan-Bing He,[1] Huan-Ling Liu,[2] and Kai-Jia Sun[3,5]



# Active learning to map out unphysical EoS

$$(\mu_{BC}, \alpha_{\rm diff}, w, \rho) \mapsto P(T, \mu_B) \mapsto \{\text{acceptable, unstable, acausal}\}.$$

4 parameters from 3D Ising model

QCD EoS

Lables for classification

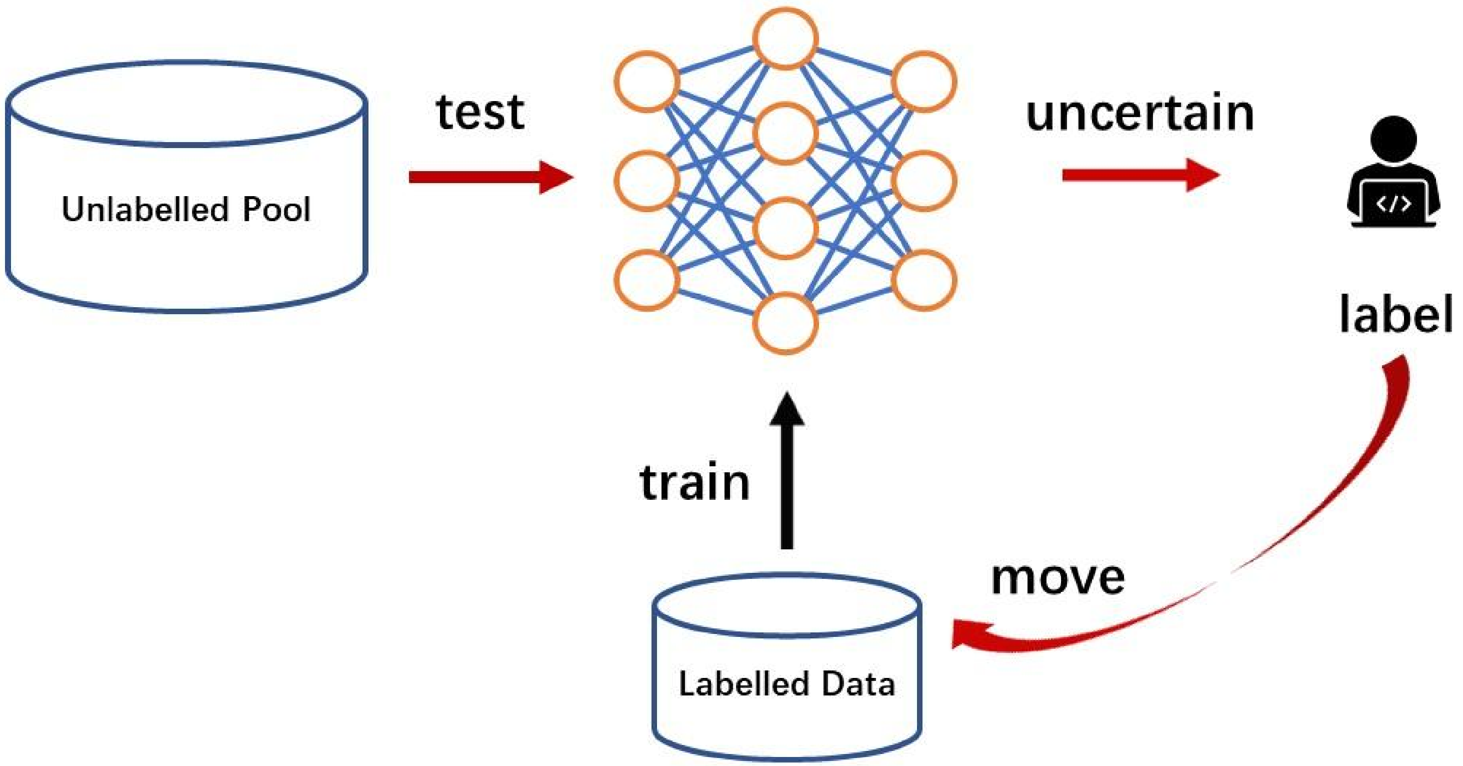
$\boxed{\text{Acceptable = Stable + Causal}}$

$$P, s, \varepsilon, n_B, \chi_2^B, \left(\frac{\partial S}{\partial T}\right)_{n_B} > 0\ ,$$

$$0 \le c_s^2 \le 1.$$

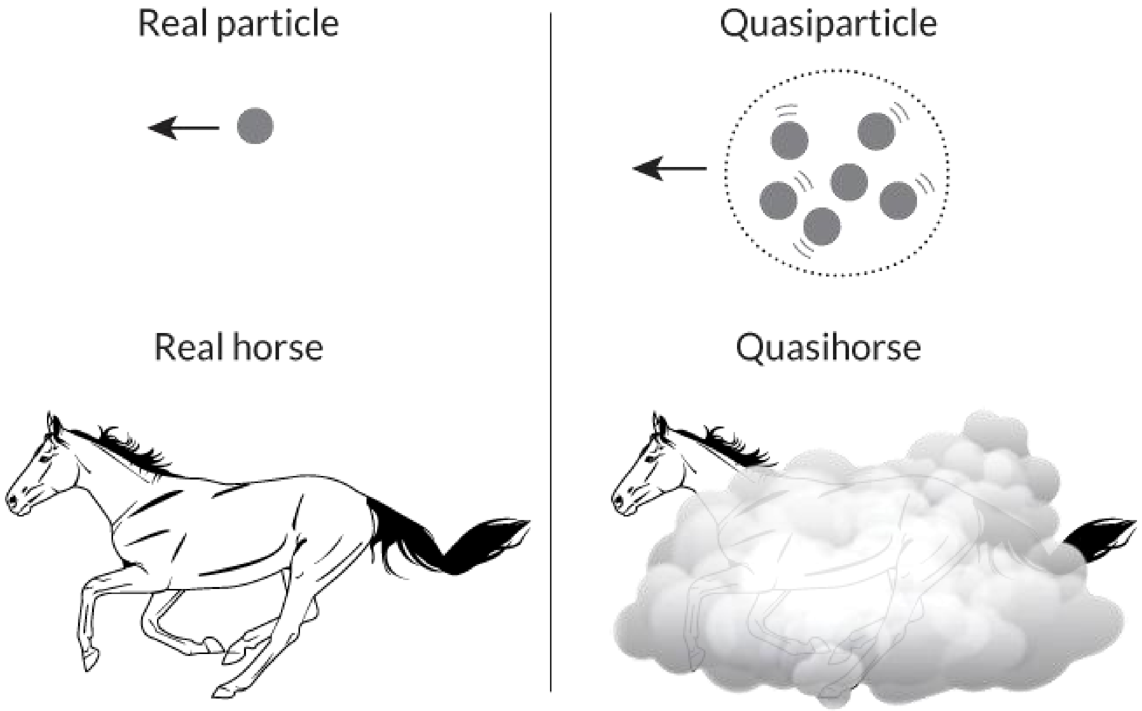
D. Mroczek, M. Hjorth-Jensen, J. Noronha-Hostler, P. Parotto, C. Ratti, and R. Vilalta, PRC 107, 054911

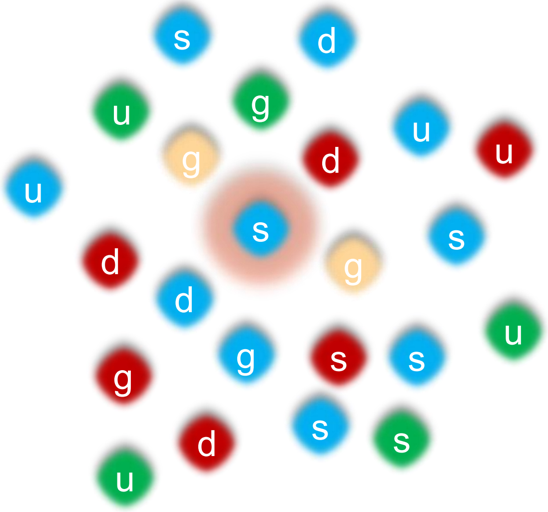
# Active learning procedure




D. Mroczek, M. Hjorth-Jensen, J. Noronha-Hostler, P. Parotto, C. Ratti, and R. Vilalta, PRC 107, 054911

# Quasi particle picture of QCD EoS

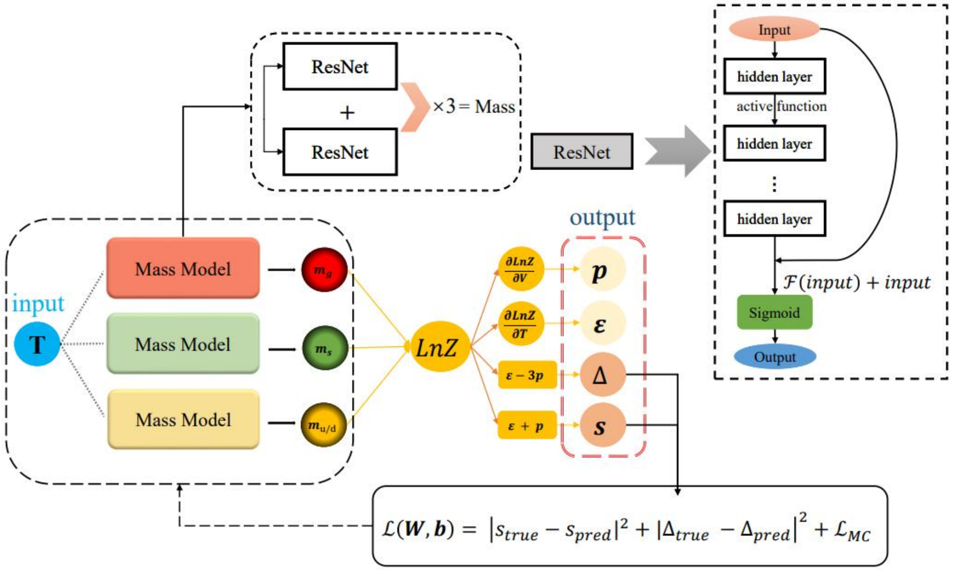



screened, dressed, regularized, quasi particle

# DL For Quasi Particle Mass

FuPeng Li, HL Lu, LG Pang, GY Qin, PLB 2023




$$\ln Z(T) = \ln Z_g(T) + \ln Z_{u,d}(T) + \ln Z_s(T),$$

Fermi-Dirac distributions,

$$\ln Z_g(T) = -\frac{16V}{2\pi^2}\int_0^\infty p^2 dp \ln\left[1 - \exp\left(-\frac{1}{T}\sqrt{p^2 + m_g^2(T)}\right)\right], \quad (2)$$

$$\ln Z_{q_i}(T) = +\frac{12V}{2\pi^2}\int_0^\infty p^2 dp \ln\left[1 + \exp\left(-\frac{1}{T}\sqrt{p^2 + m_{q_i}^2(T)}\right)\right], \quad (3)$$

quarks, $m_s(T, \theta_2)$ for strange quark and $m_g(T, \theta_3)$ for gluons, where $\theta_1$, $\theta_2$ and $\theta_3$ are the parameters in DNN shown in Fig. 1.

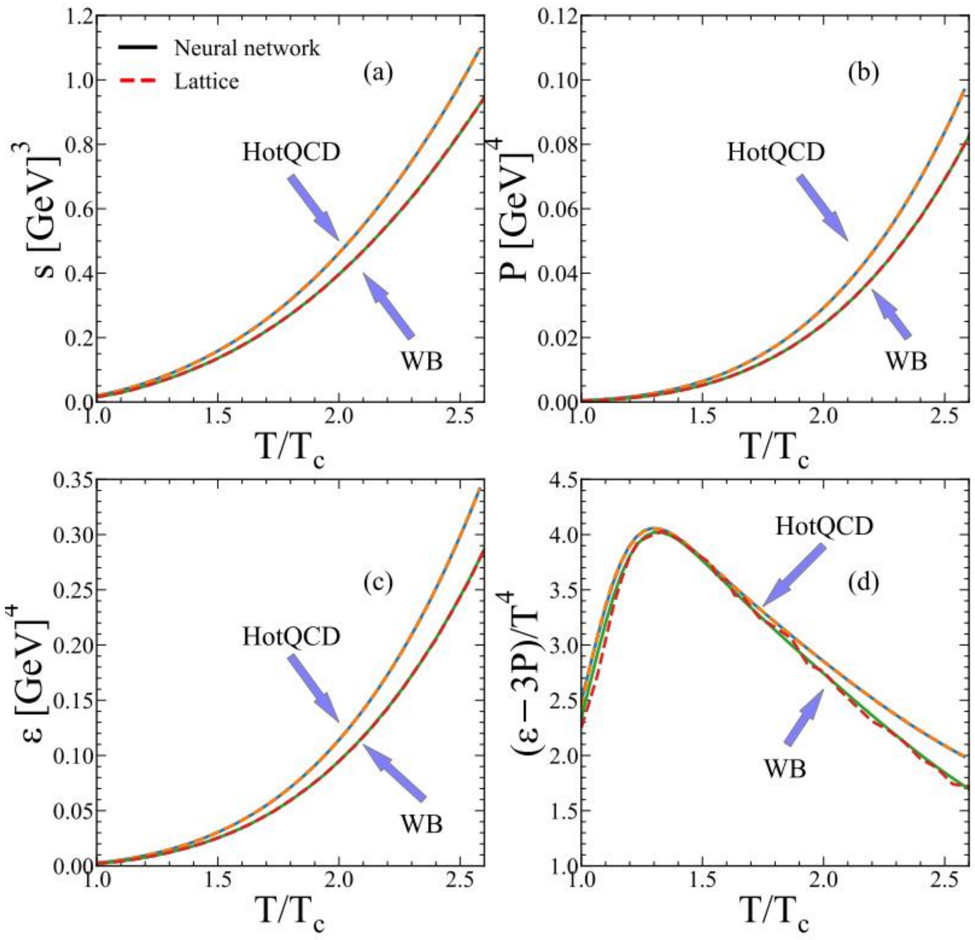
The resulting pressure and energy density are computed using the following statistical formulae,

$$P(T) = T\left(\frac{\partial \ln Z(T)}{\partial V}\right)_T, \quad (5)$$

$$\epsilon(T) = \frac{T^2}{V}\left(\frac{\partial \ln Z(T)}{\partial T}\right)_V, \quad (6)$$

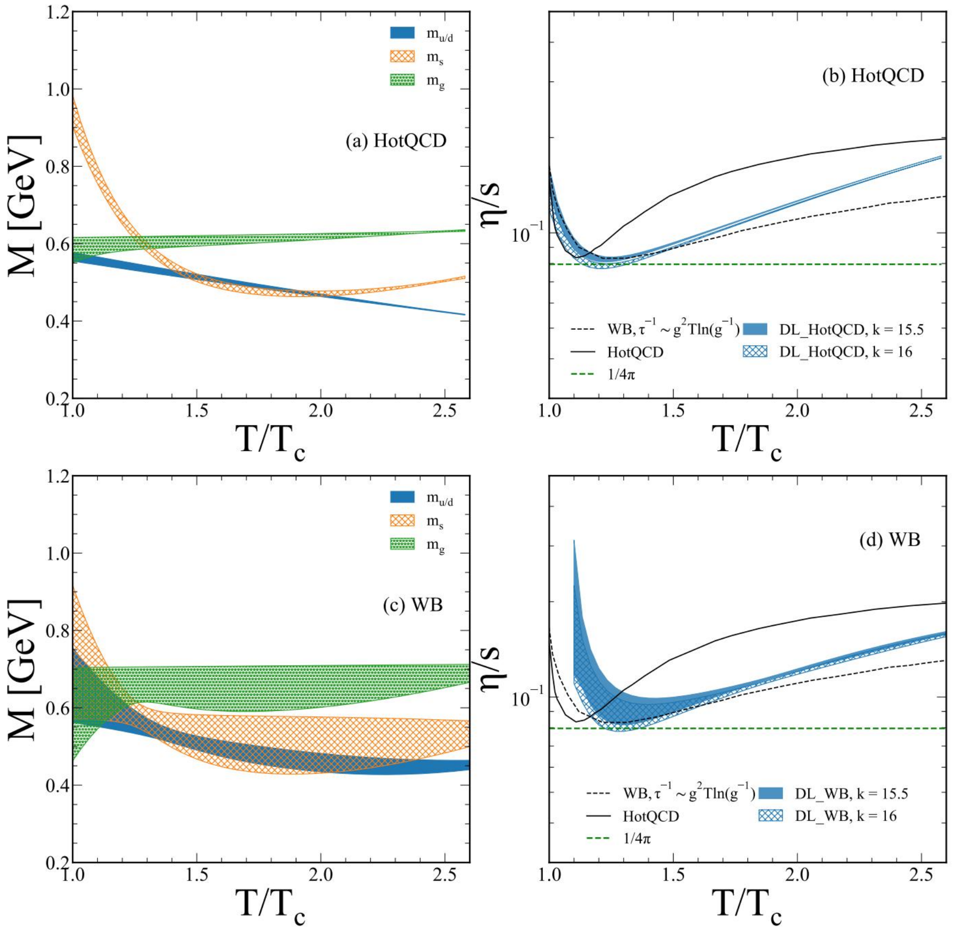
# The learned quasi parton mass

**EoS vs Lattice QCD**



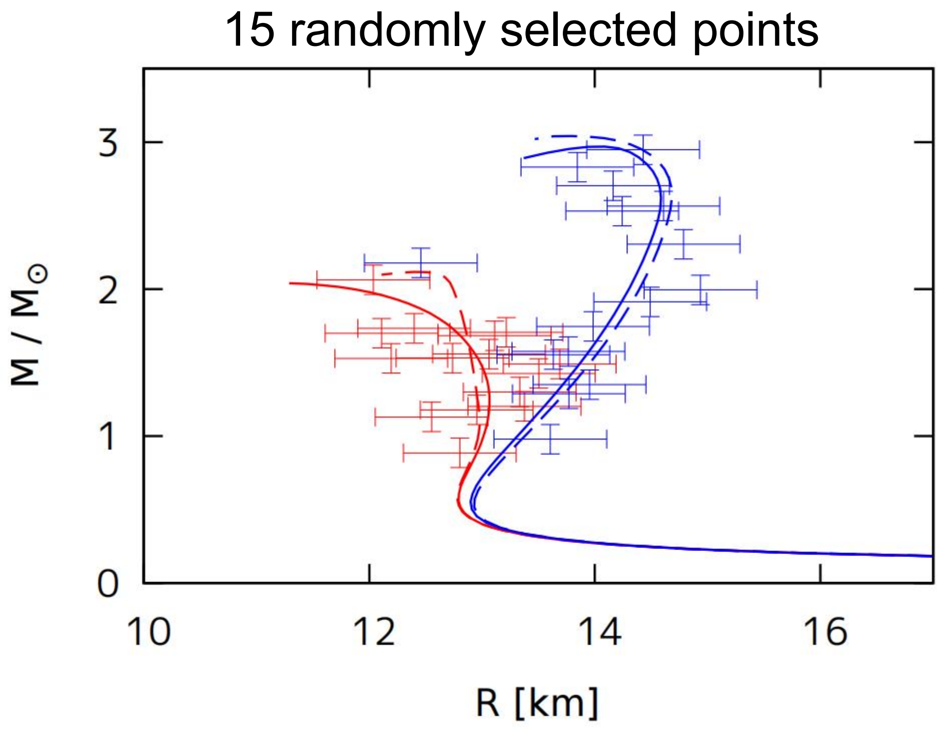

**Learned Mass**

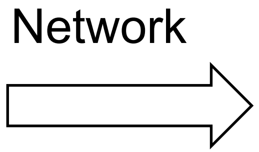
$\eta/s$

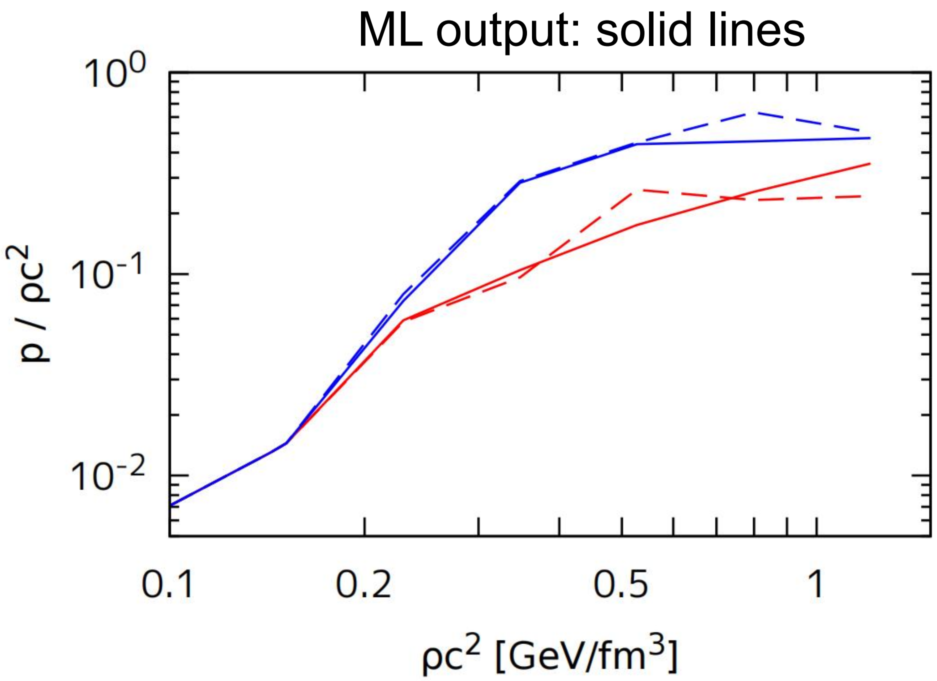



FuPeng Li, HL Lu, LG Pang, GY Qin, PLB 2023

# TOV equation and nuclear EoS from DL



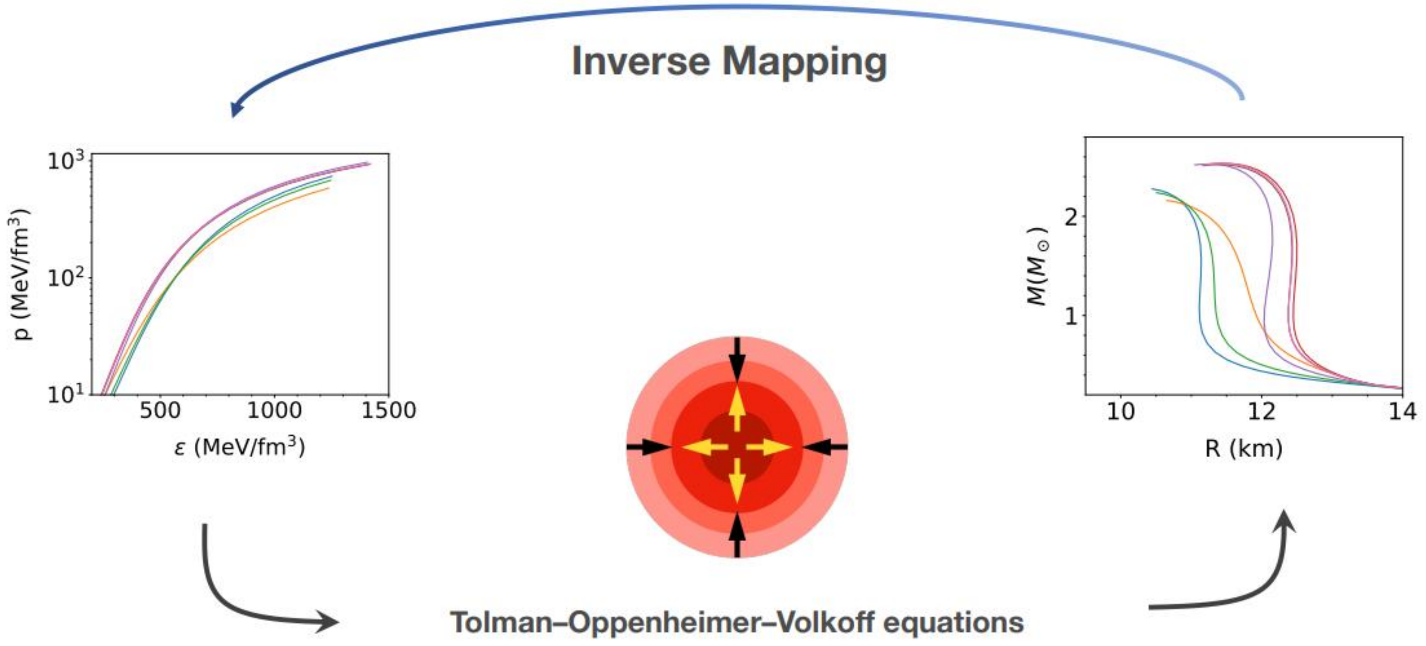




Yuki Fujimoto, Kenji Fukushima, and Koichi Murase, PRD 98 (2018) 2, 023019

# TOV Equation and Nuclear EoS from DL

$$\frac{\mathrm{d}p}{\mathrm{d}r} = -G\frac{m(r)\varepsilon(r)}{r^2}\left(1+\frac{p(r)}{\varepsilon(r)}\right)\left(1+\frac{4\pi r^3 p(r)}{m(r)}\right)\left(1-\frac{2Gm(r)}{r}\right)^{-1},$$
$$\frac{\mathrm{d}m}{\mathrm{d}r} = 4\pi r^2\varepsilon\,,$$



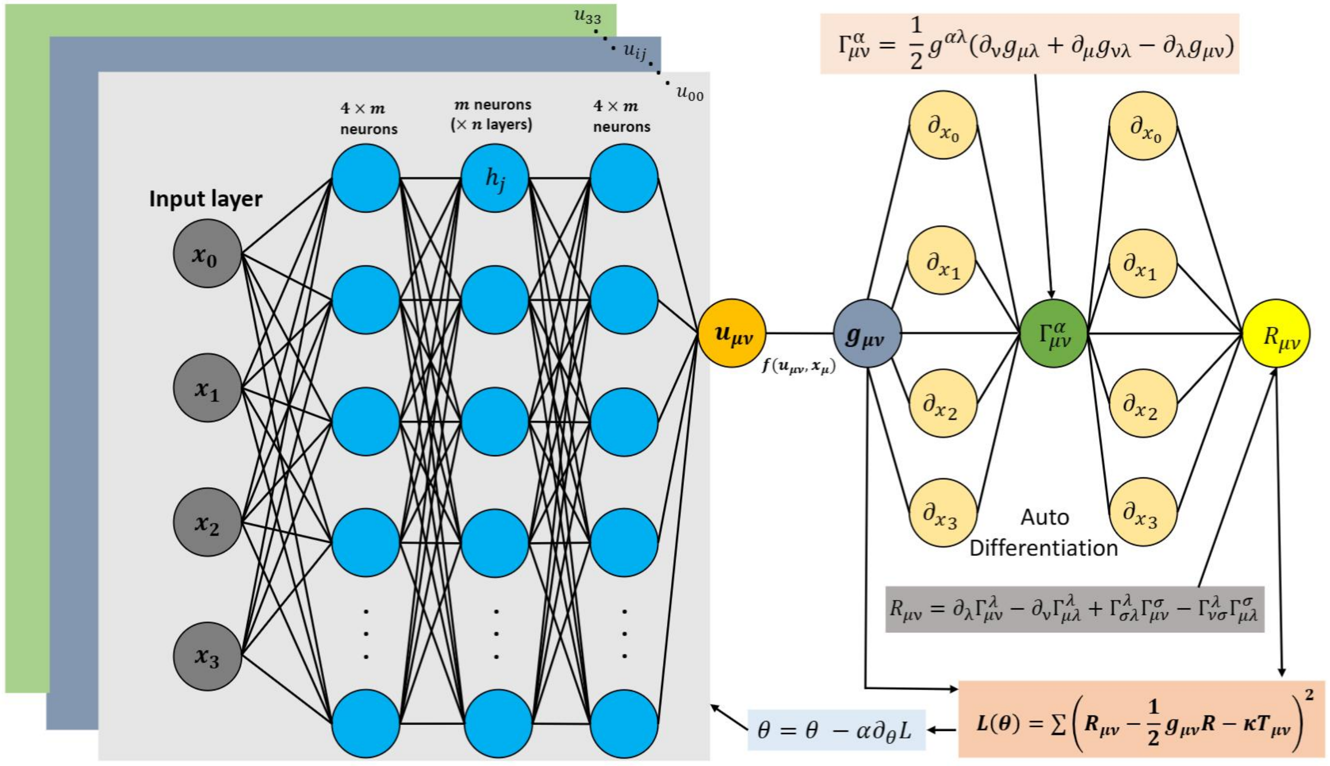

S. Soma, L. Wang, S. Shi, H. Stöcker, K. Zhou, PRD 107, (2023) 083028

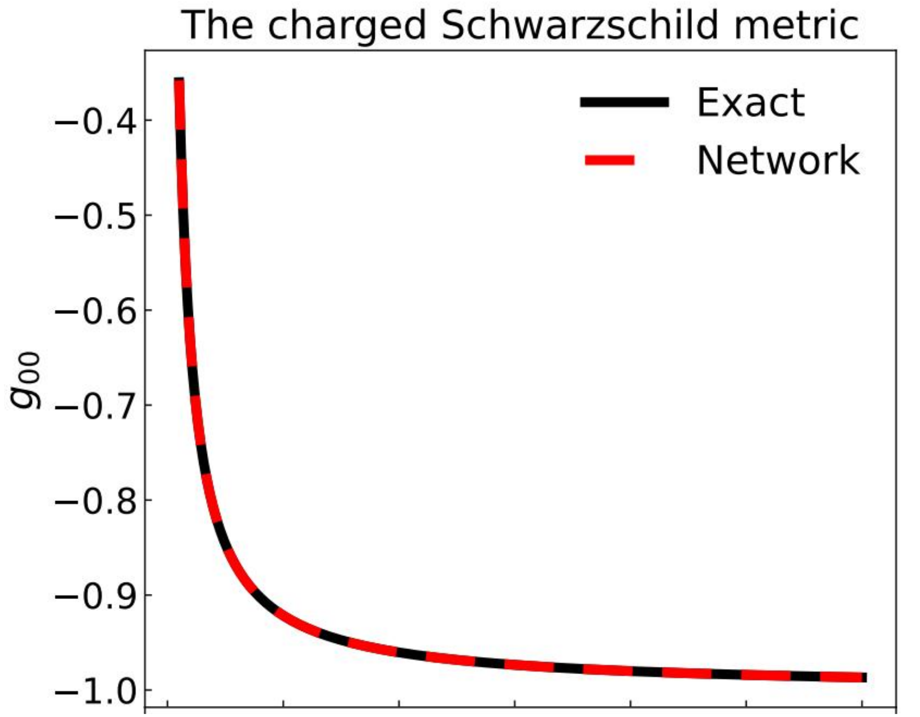
Solving Einstein equations using deep learning

Zhi-Han Li[1], Chen-Qi Li[1], Long-Gang Pang[1a]
[1] *Key Laboratory of Quark & Lepton Physics (MOE) and Institute of Particle Physics, Central China Normal University, Wuhan 430079, China*

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \kappa T_{\mu\nu}$$

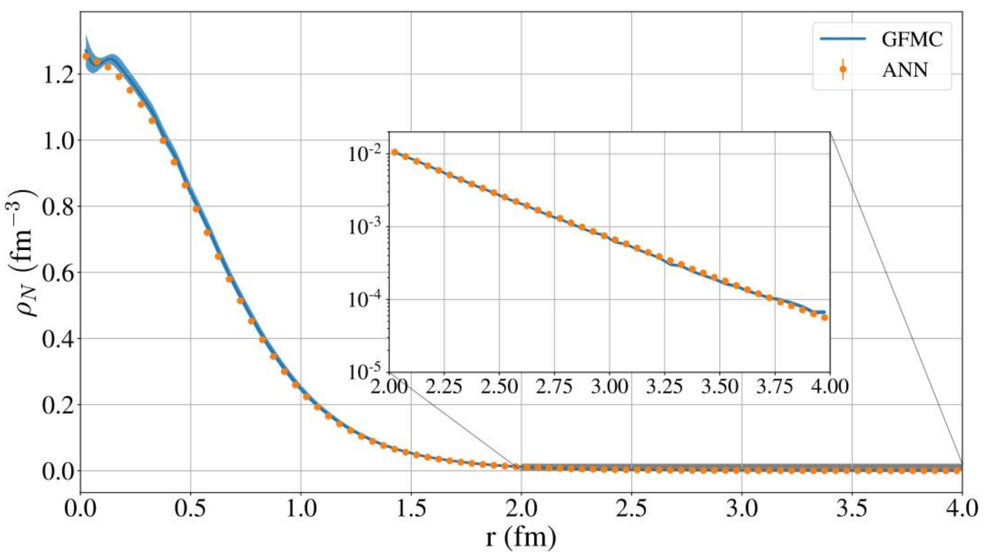
$$L(\theta) = \frac{1}{N} \sum_{i=0}^{N} \left(R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R - \kappa T_{\mu\nu}\right)^2$$



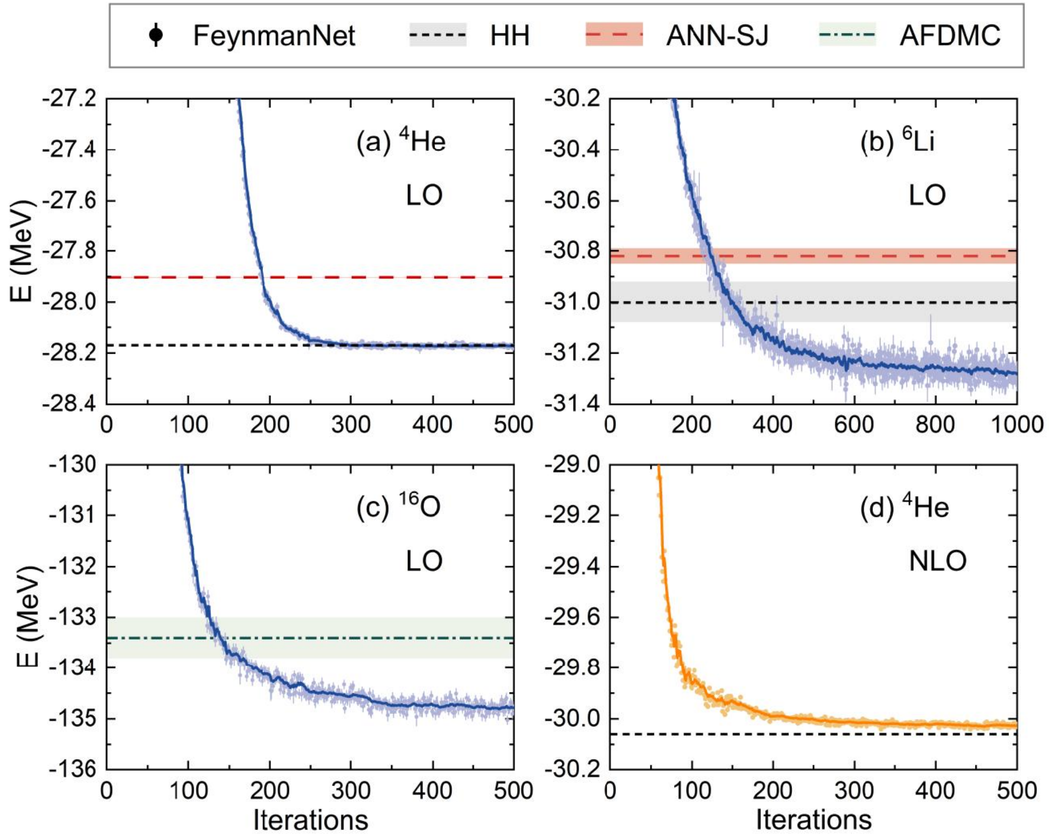



# Nuclear many body problem

$$\hat{H} = \sum_{i=1}^{A} \frac{-\nabla_i^2}{2m_N} + \sum_{i<j} v_{ij} + \sum_{i<j<k} V_{ijk},$$




Adams et al., 2021



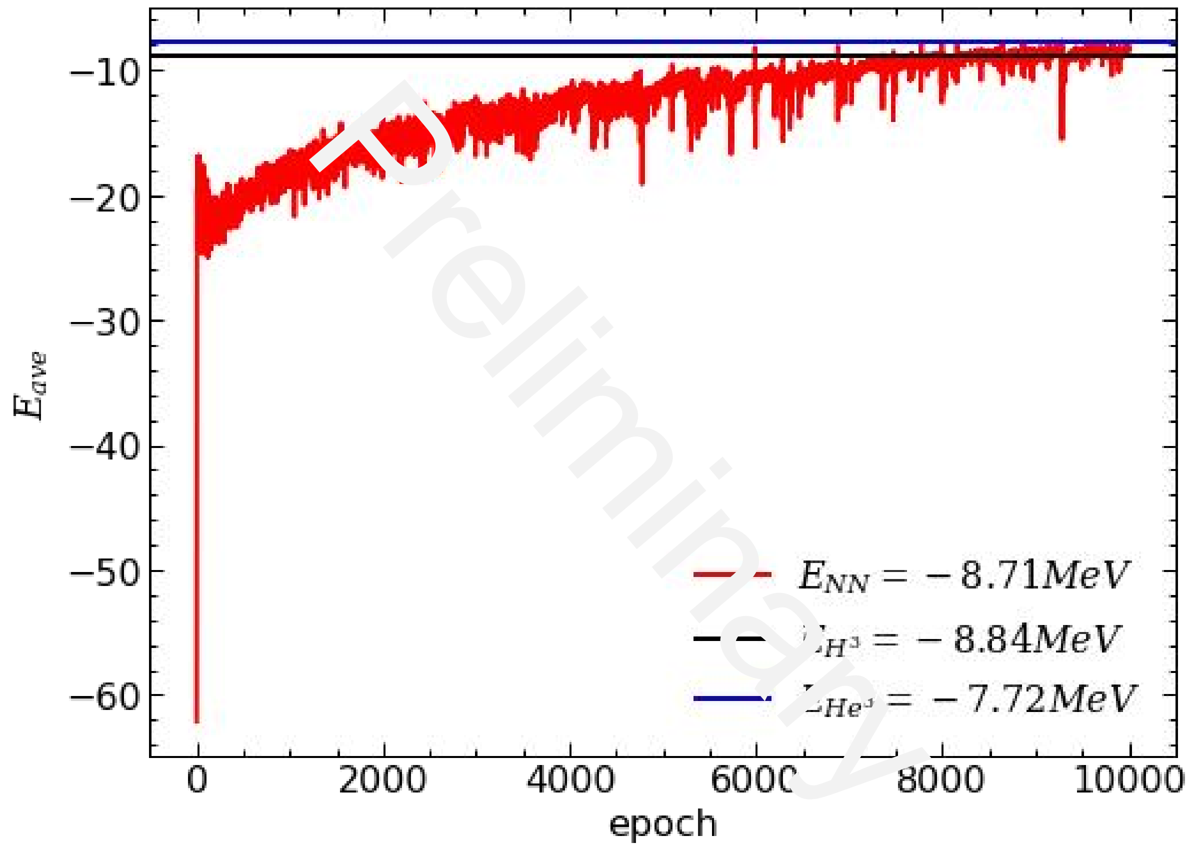

Y.L. Yang, P.W. Zhao, PRC 2023

# Using PINN

- 使用人工神经网络表示波函数
- 使用自动微分计算动能项
- 思路：优化目标

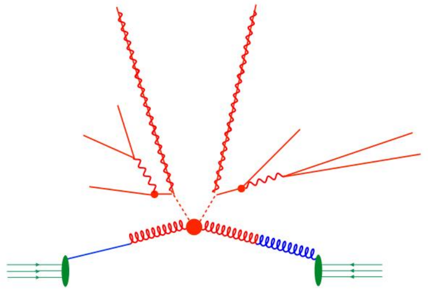
$$L = \lambda_1 || H|\Psi\rangle - E|\Psi\rangle || + \lambda_2 e^{E}$$

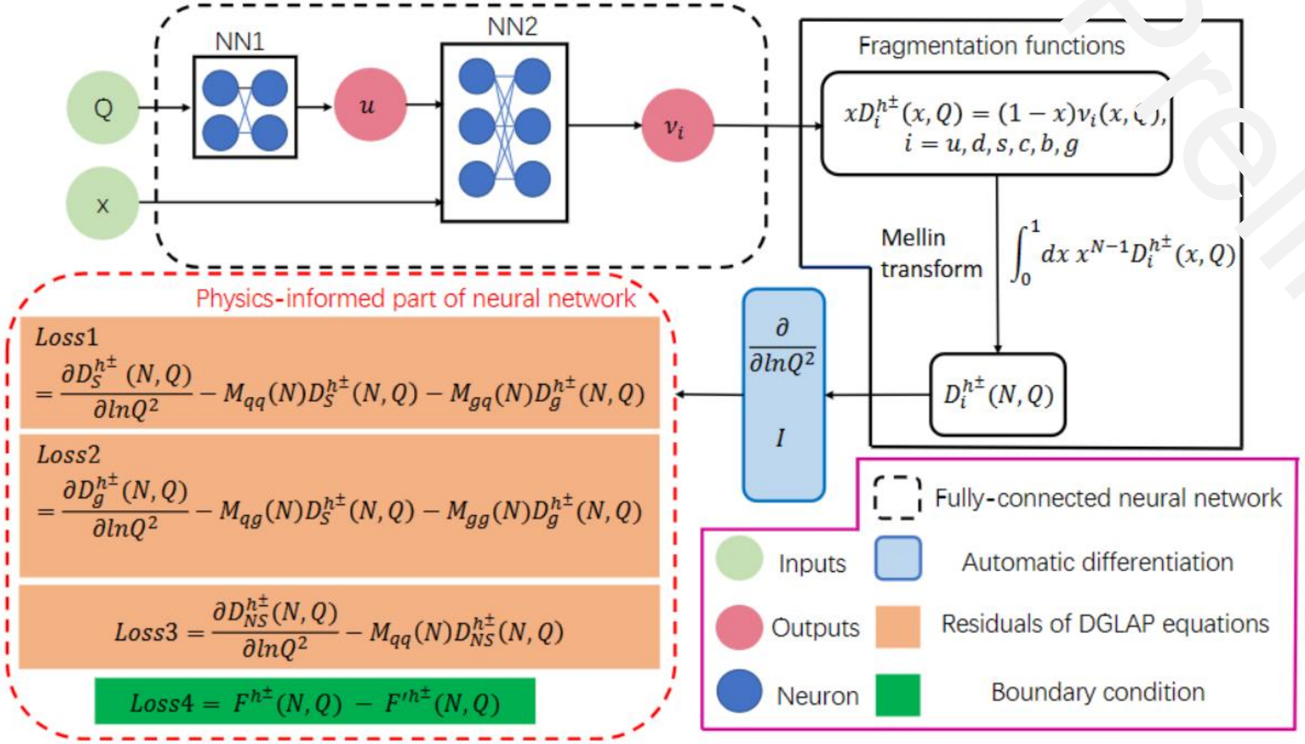
With FuPeng Li and Xin-Nian Wang

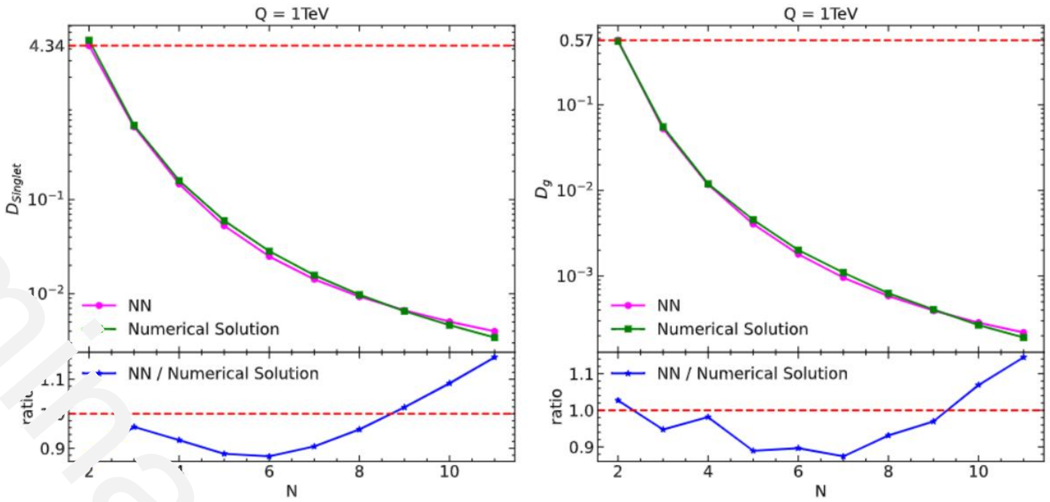


# Auto-Diff for parton fragmentation function

$$\frac{\partial}{\partial \ln Q^2} D_i^{h^\pm}(x,Q) = \frac{\alpha_s(Q)}{2\pi} P_{ij}(x,\alpha_s) \otimes D_j^{h^\pm}(x,Q)$$





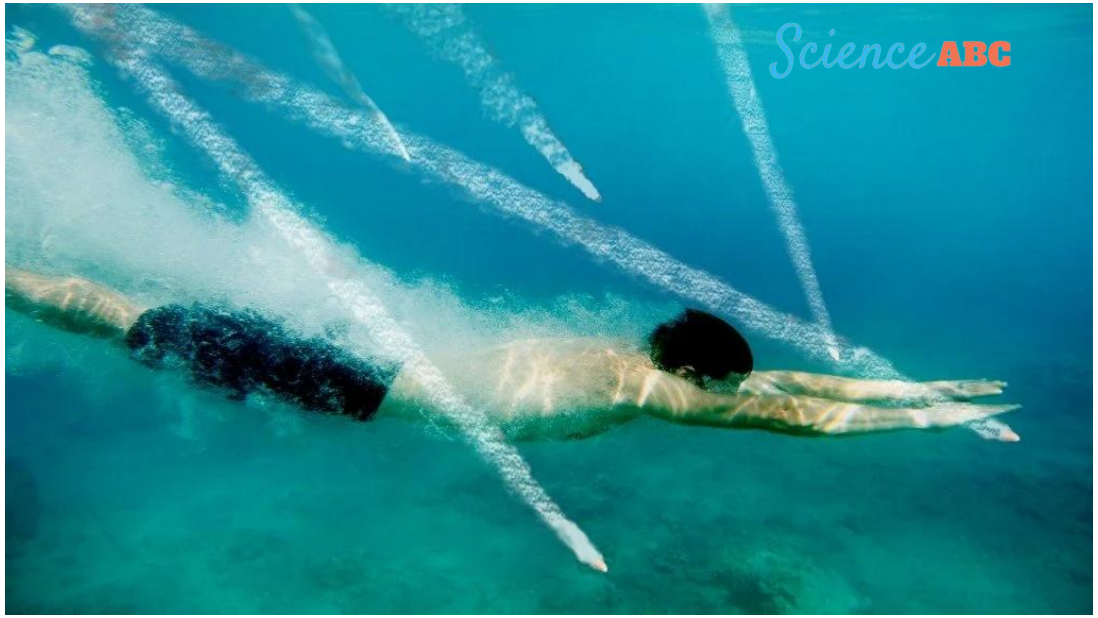



1. 结果与传统数值解符合
2. 自动满足 DGLAP 演化方程

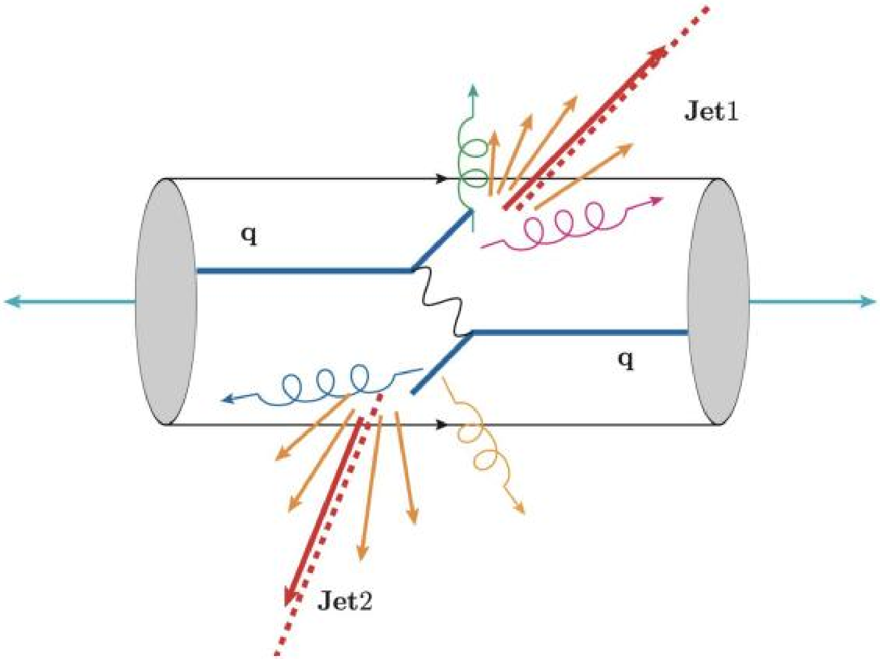
SW Dai, FP Li, LG Pang, XN Wang,
BW Zhang, HZ Zhang, in preparation

# Jet quecnhing

Can Being Underwater Protect You From Bullets?

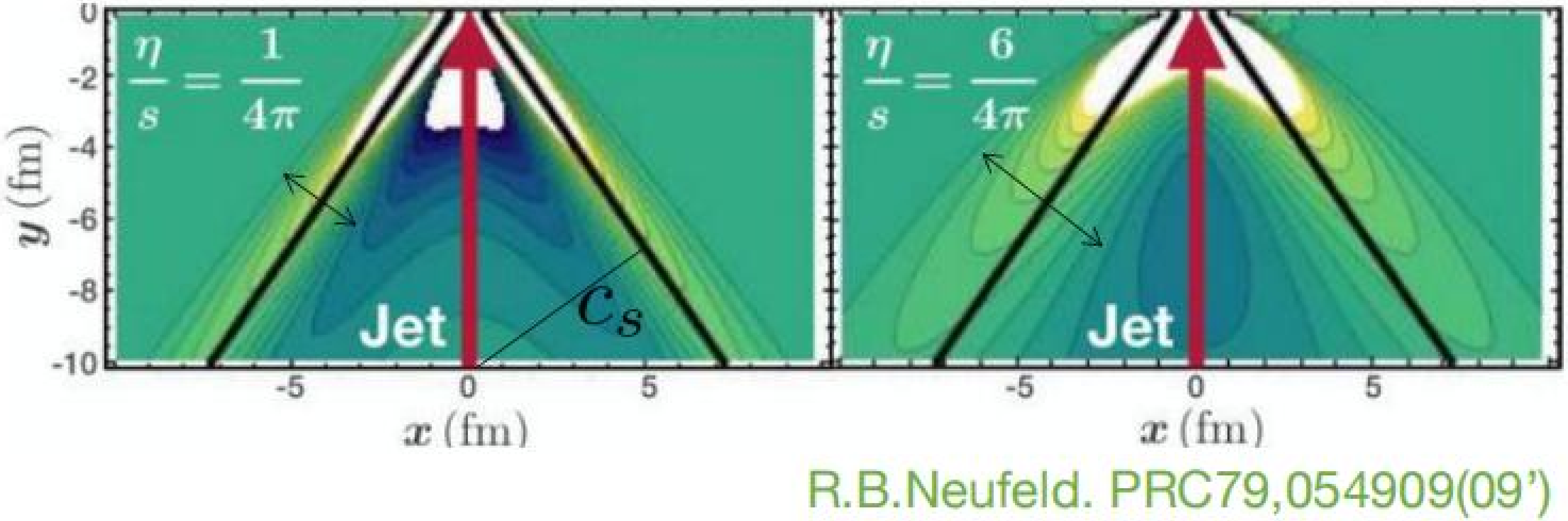



> *If the bullet is shot from an angle of 30 Degrees, then being underwater in the range of 3-5 feet (0.9-1.5 meters) can ensure safety from most guns.*




Jet quenching in hot QGP

# The nuclear EoS and Mach Cone

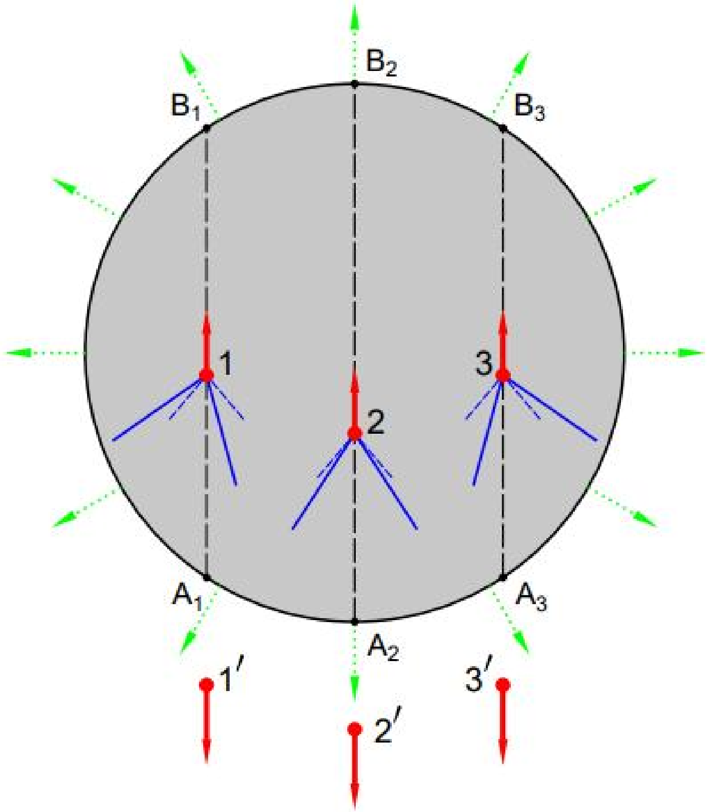



Nuclear EoS： $c_s^2 = \frac{dP}{d\epsilon} = \sin^2\theta$

Shear Viscosity： width of the shock wave

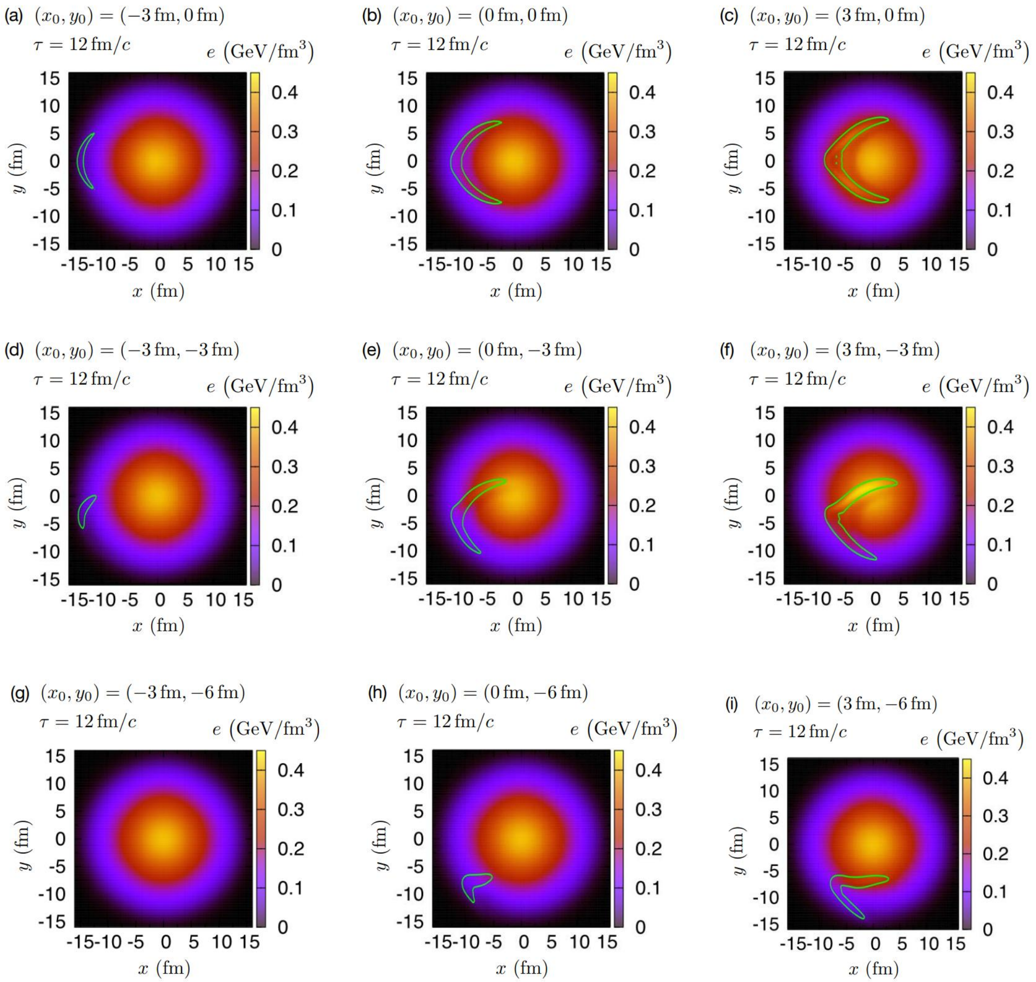
# Difficulties in looking for Mach Cones in HIC

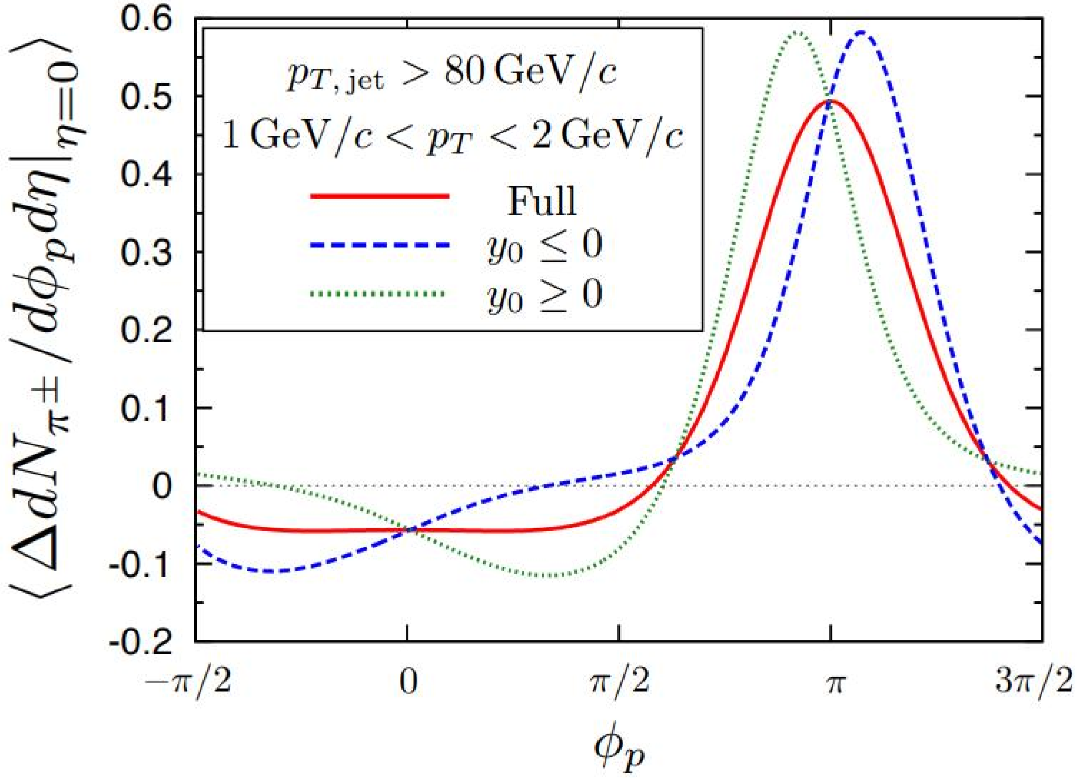
- Random production locations and propagating directions relative to collective flow
- Tilted by different path length and collective flow




L.M. Satarov, H. Stoecker, I.N. Mishustin, PLB 627 (2005) 64-70

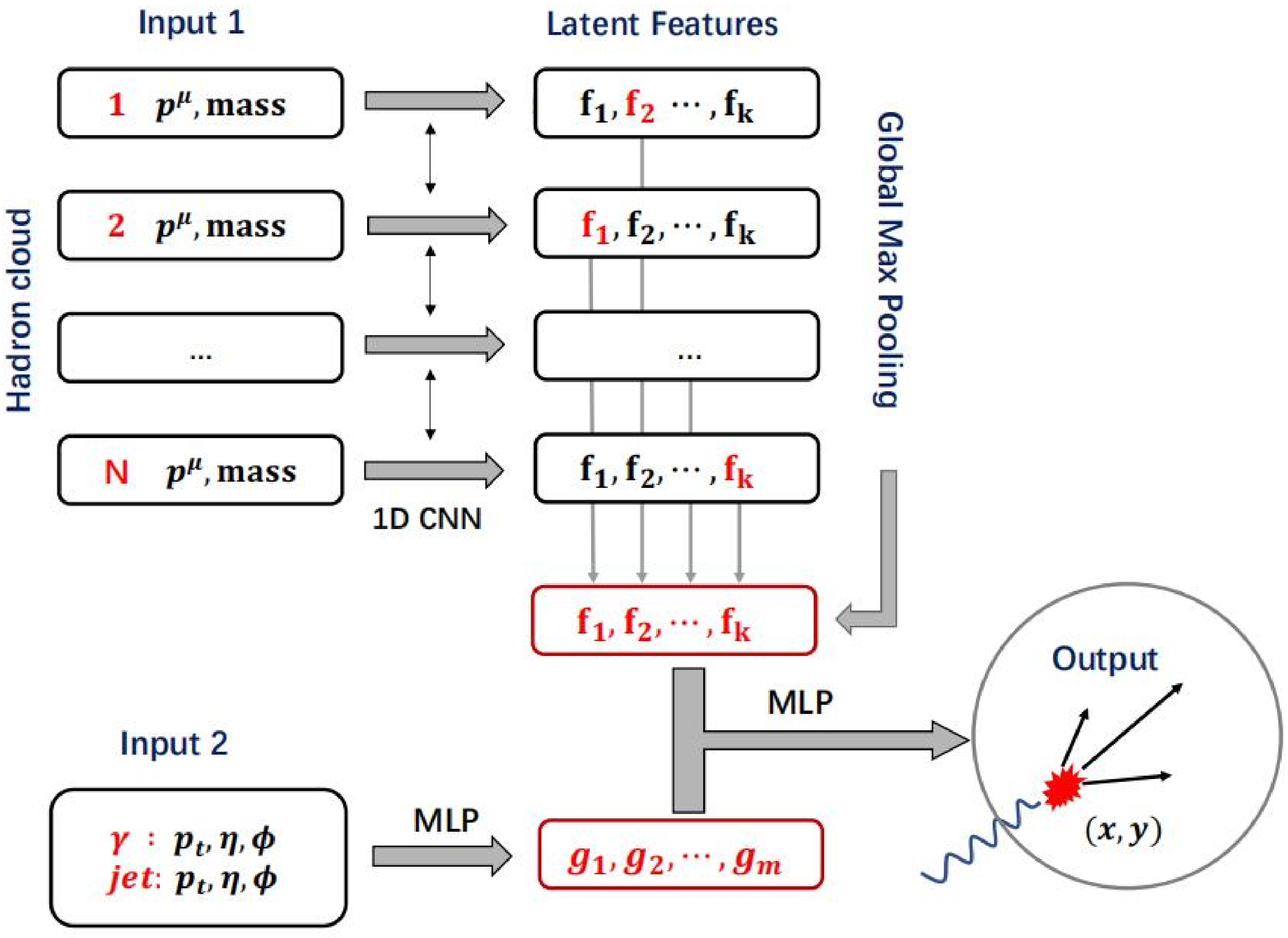
# If it is possible to locate the initial jets




Y Tachibana, T Hirano, PRC 93 (2016) 5, 054907

# DL assisted jet tomography (gamma-jet)




| $ij \to kl$ | $\lvert M\rvert^2_{ij\to kl}$ | |
|---|---|---|
| $gg \to gg$ | $\frac{9}{2} g_s^4 \left(3 - \frac{ut}{s^2} - \frac{us}{t^2} - \frac{st}{u^2}\right)$ | (A-1) |
| $gg \to q\bar{q}$ | $\frac{3}{8} g_s^4 \left(\frac{4}{9}\frac{t^2+u^2}{tu} - \frac{t^2+u^2}{s^2}\right)$ | (A-2) |
| $gq \to gq$<br>$g\bar{q} \to g\bar{q}$ | $g_s^4 \left(\frac{s^2+u^2}{t^2} - \frac{4}{9}\frac{s^2+u^2}{su}\right)$ | (A-3) |
| $q_i q_j \to q_i q_j$<br>$q_i \bar{q}_j \to q_i \bar{q}_j$<br>$\bar{q}_i q_j \to \bar{q}_i q_j$<br>$\bar{q}_i \bar{q}_j \to \bar{q}_i \bar{q}_j$ | $\frac{4}{9} g_s^4 \frac{s^2+u^2}{t^2}, \quad i \neq j$ | (A-4) |
| $q_i q_i \to q_i q_i$<br>$\bar{q}_i \bar{q}_i \to \bar{q}_i \bar{q}_i$ | $\frac{4}{9} g_s^4 \left(\frac{s^2+u^2}{t^2} + \frac{s^2+t^2}{u^2} - \frac{2}{3}\frac{s^2}{tu}\right)$ | (A-5) |
| $q_i \bar{q}_i \to q_j \bar{q}_j$ | $\frac{4}{9} g_s^4 \frac{t^2+u^2}{s^2}$ | (A-6) |
| $q_i \bar{q}_i \to q_i \bar{q}_i$ | $\frac{4}{9} g_s^4 \left(\frac{s^2+u^2}{t^2} + \frac{t^2+u^2}{s^2} - \frac{2}{3}\frac{u^2}{st}\right)$ | (A-7) |
| $q\bar{q} \to gg$ | $\frac{8}{3} g_s^4 \left(\frac{4}{9}\frac{t^2+u^2}{tu} - \frac{t^2+u^2}{s^2}\right)$ | (A-8) |

$$(x_i^{\mathrm{net}}, y_i^{\mathrm{net}}) = f(\{\vec{p}\}_i, \theta),$$

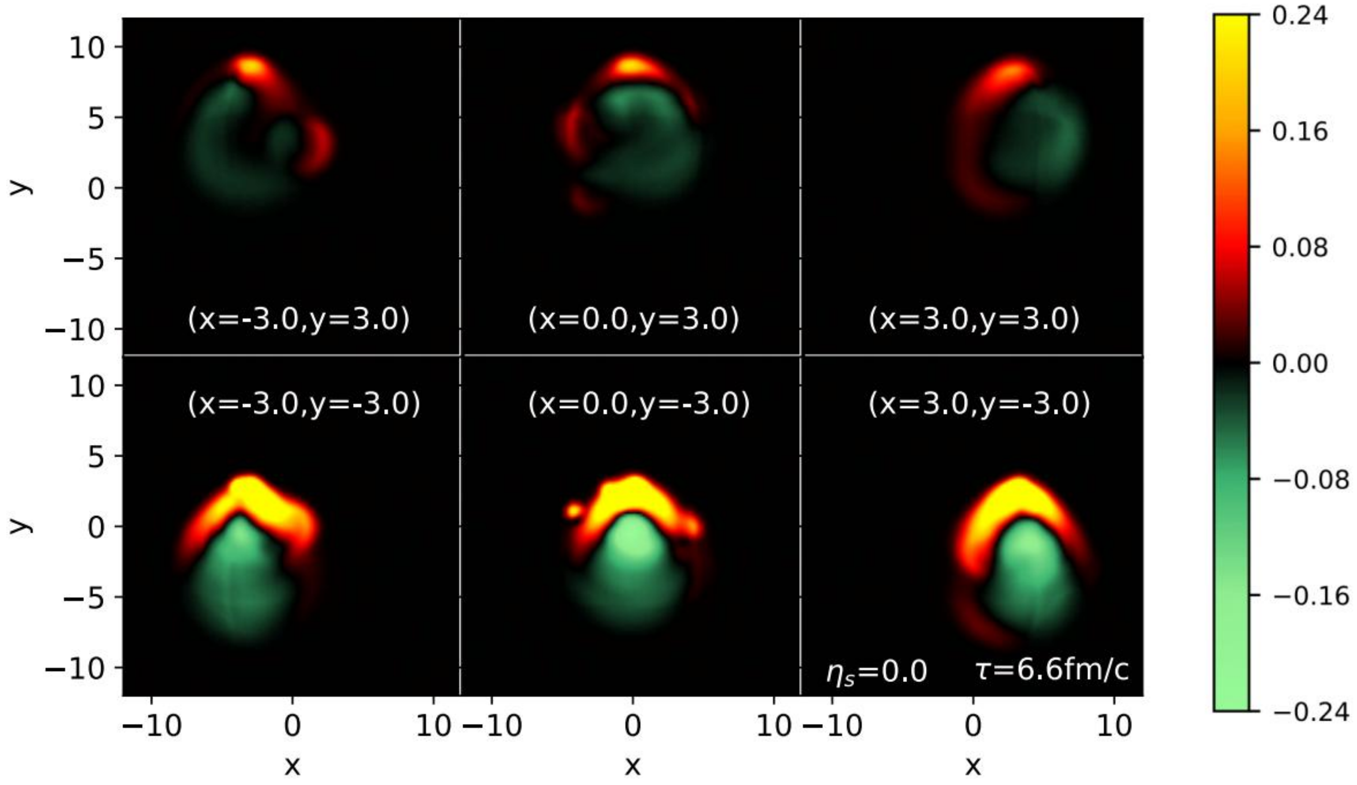
Z Yang, YY He, W Chen, WY Ke, LG Pang, XN Wang, EPJC 83 (2023) 7, 652

# Training data: CoLBT(LBT + CLVisc)

$$p\partial f(p) = -C(p) \quad (p\cdot u > p^0_{cut})$$

$$\partial_\mu T^{\mu\nu}(x) = j^\nu(x)$$

$$j^\nu = \sum_i p_i^\nu \delta^{(4)}(x-x_i)\theta(p^0_{cut} - p\cdot u)$$




**LBT: YY He, T Luo, XN Wang, Y Zhu,**
PRC 91 (2015) 054908, PRC 97 (2018) 1, 019902

**CLVisc:**
**LG Pang, Q Wang, XN Wang, PRC 86 (2012) 024911**
**LG Pang, H Petersen, XN Wang,** PRC 97 (2018) 6, 064918
**XY Wu, GY Qin, LG Pang, XN Wang,**PRC 105 (2022) 3, 034909

**CoLBT**:

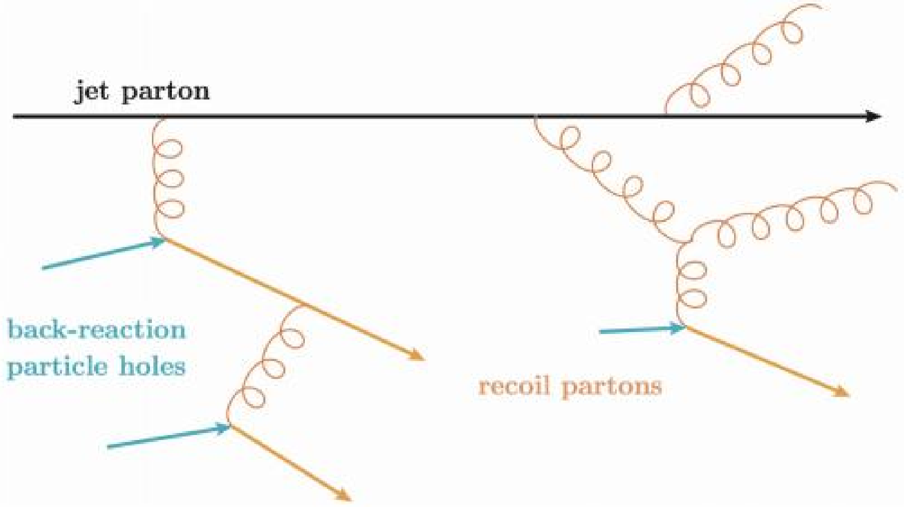
**W Chen, T Luo, SS Cao, LG Pang, XN Wang,**
PLB 777 (2018) 86-90

# LBT: Linear Boltzmann Transport

$$p_1 \partial f_1 = -\int dp_2 dp_3 dp_4 (f_1 f_2 - f_3 f_4) \left|M_{12\to 34}\right|^2 (2\pi)^4 \delta^4 (\sum_i p^i) + inelastic$$

Medium-induced gluon(HT):

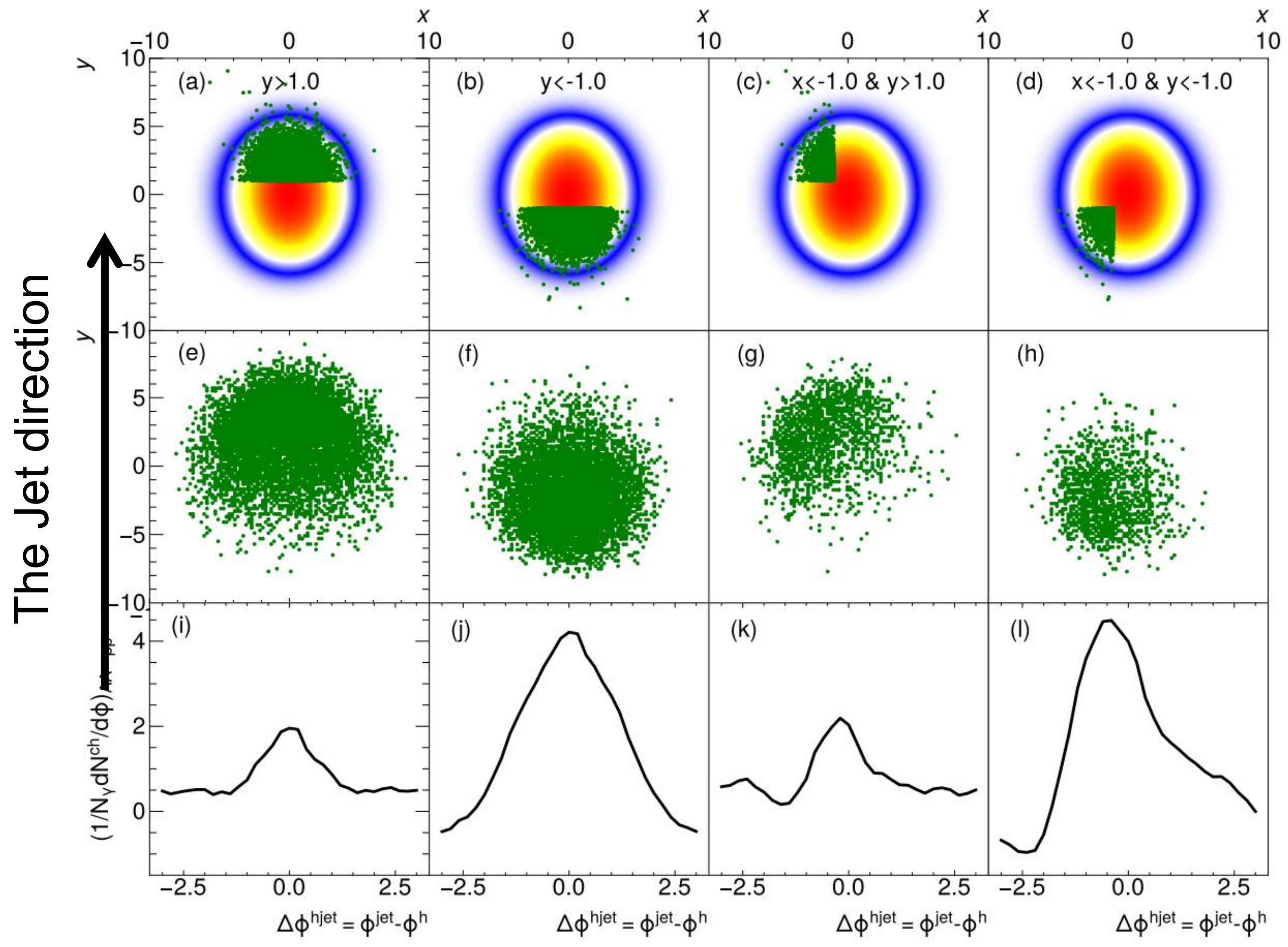
$$\frac{dN_g}{dz d^2 k_\perp dt} \approx \frac{2C_A \alpha_s}{\pi k_\perp^4} P(z) \hat{q} (\hat{p} \cdot u) sin^2 \frac{k_\perp^2 (t - t_0)}{4z(1-z)E}$$




Tracked partons:
- Jet shower partons
- Thermal recoil partons
- Radiated gluons
- Negative partons(Back reaction induced by energy-momentum conservation)

YY He, T Luo, XN Wang, Y Zhu, PRC 91 (2015) 054908, PRC 97 (2018) 1, 019902

# DL assisted jet tomography



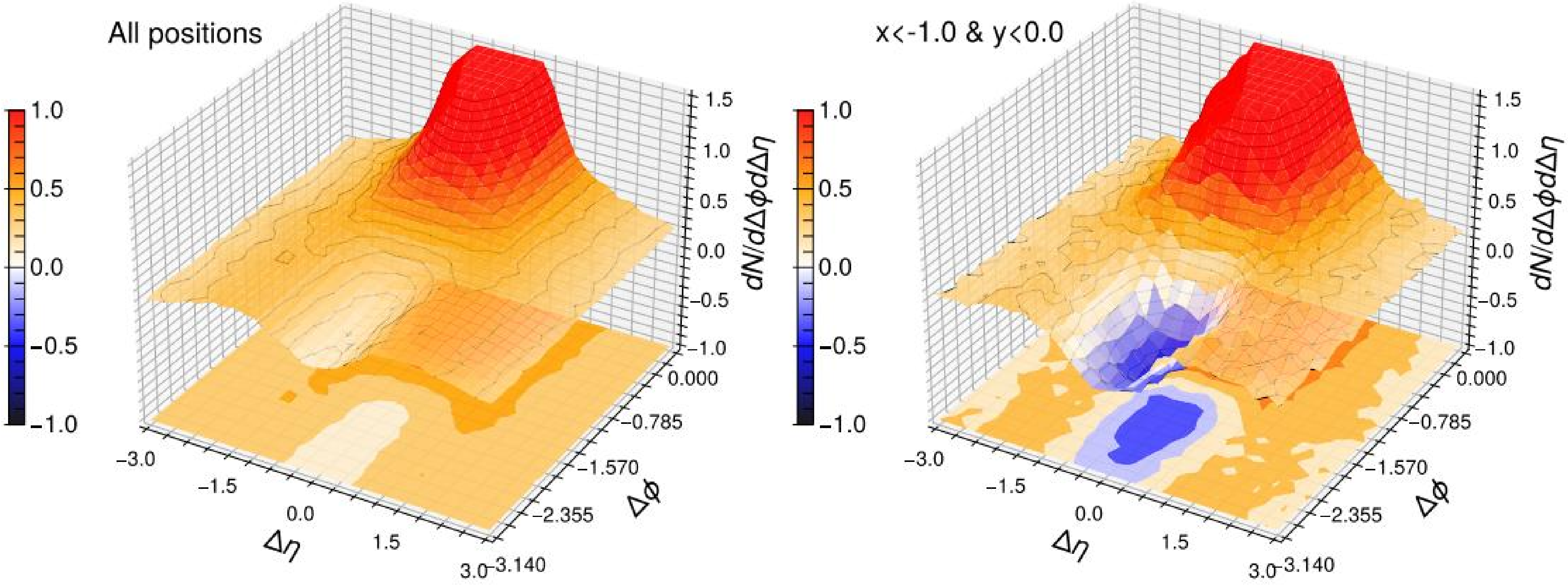

Network predictions

True locations

Jet hadron correlation for selected events whose locations are constrained to specific regions using DL assisted jet tomography

Z Yang, YY He, W Chen, WY Ke, LG Pang, XN Wang, EPJC 83 (2023) 7, 652

# Enhance the Diffusion Wake signal




Z Yang, YY He, W Chen, WY Ke, LG Pang, XN Wang, EPJC 83 (2023) 7, 652

Z Yang, T Luo, W Chen, LG Pang, XN Wang, PRL 130 (2023) 5, 052301

# Summary

- Studying physics using HIC is a typical inverse problem
- DL builds nonlinear maps between final state observations and physical properties of nuclear matter
- Deep neural networks are widely used as varational functions that are learned through training
- The derivatives of the varational function can be computed with machine precision using auto-diff