



# High-energy nuclear physics meets machine learning

Wan-Bing He<sup>1,2</sup> · Yu-Gang Ma<sup>1,2</sup> · Long-Gang Pang<sup>3</sup> · Hui-Chao Song<sup>4</sup> · Kai Zhou<sup>5</sup>

Received: 10 March 2023 / Revised: 13 April 2023 / Accepted: 18 April 2023 / Published online: 21 June 2023  
© The Author(s) 2023

## Abstract

Although seemingly disparate, high-energy nuclear physics (HENP) and machine learning (ML) have begun to merge in the last few years, yielding interesting results. It is worthy to raise the profile of utilizing this novel mindset from ML in HENP, to help interested readers see the breadth of activities around this intersection. The aim of this mini-review is to inform the community of the current status and present an overview of the application of ML to HENP. From different aspects and using examples, we examine how scientific questions involving HENP can be answered using ML.

**Keywords** Heavy-ion collisions · Machine learning · Initial state · Bulk properties · Medium effects · Hard probes · Observables

## 1 Introduction

Machine learning (ML) has a long history of development and application spanning several decades. It is a rapidly growing field of modern science and endows computers with the ability to learn and make predictions from data without explicit programming. It falls under the umbrella of artificial intelligence (AI) and is closely related to statistical inference and pattern recognition. Recently, ML technologies have experienced a revival and gained popularity—particularly after AlphaGo from DeepMind defeated the human champion in the game of Go. This resurgence can be attributed to the advancement of algorithms, the increasing availability of

powerful computational hardware such as graphics processing units (GPUs), and the abundance of data.

Nuclear physics seeks to understand the nature of nuclear matter, including its fundamental constituents and collective behavior under different conditions, as well as the fundamental interactions that govern them. Traditional nuclear physics—particularly for energies below approximately 1 GeV/nucleon—focuses on nuclear structures and reactions, where the degree of freedom is the nucleon. However, in high-energy nuclear physics (HENP), the degree of freedom includes and is often dominated by quarks and gluons. Theoretical calculations and experiments or observations with large scientific infrastructures play a leading role but are reaching unprecedented complexity and scale. In the context of HENP—particularly nuclear collisions—researchers are already at the forefront of *Big Data analysis*. The detectors

This work was supported in part by the National Natural Science Foundation of China under contract Nos. 11890714 and 12147101 (Ma), 12075098 (Pang), and 12247107 and 12075007 (Song); the Germany BMBF under the ErUM-Data project (Zhou); and the Guangdong Major Project of Basic and Applied Basic Research No. 2020B0301030008 (Ma).

✉ Kai Zhou  
zhou@fias.uni-frankfurt.de

Wan-Bing He  
hewanbing@fudan.edu.cn

Yu-Gang Ma  
mayugang@fudan.edu.cn

Long-Gang Pang  
lpgang@mail.ccnu.edu.cn

Hui-Chao Song  
huichaosong@pku.edu.cn

<sup>1</sup> Key Laboratory of Nuclear Physics and Ion-beam Application (MOE), Institute of Modern Physics, Fudan University, Shanghai 200433, China

<sup>2</sup> Shanghai Research Center for Theoretical Nuclear Physics, NSFC and Fudan University, Shanghai 200438, China

<sup>3</sup> Institute of Particle Physics and Key Laboratory of Quark and Lepton Physics (MOE), Central China Normal University, Wuhan 430079, China

<sup>4</sup> School of Physics and Center for High Energy Physics, Peking University, Beijing 100871, China

<sup>5</sup> Frankfurt Institute for Advanced Studies (FIAS), 60438 Frankfurt am Main, Germany

used in high-energy nuclear collisions, such as the Relativistic Heavy Ion Collider (RHIC) and the Large Hadron Collider (LHC), can easily produce petabytes of raw data per year. A major challenge is to make sense of the vast amounts of data generated in experiments or simulated according to theory. These data are often highly complex and difficult to interpret. It is a daunting task to analyze this sheer volume of data using traditional methods of physics research. Therefore, efficient computational methods are urgently needed to facilitate physics explorations in these computational and data-intensive research areas.

One of the primary physical goals of HENP is to understand quantum chromodynamics (QCD) matter under extreme conditions. It is expected that at extremely high temperatures and/or high densities, nuclear matter, which is governed by the QCD dictated strong interaction, will turn into a deconfined quark–gluon plasma (QGP) state, which is with elementary particles—quarks and gluons—to be their basic degrees of freedom. The formation and properties of this new state of matter, as well as its transition to normal nuclear matter, are widely studied, but there are open questions in HENP. This deconfined QGP state was believed to exist in the early universe, a few microseconds after the Big Bang. Another way to study the QGP state is in terms of neutron stars (or binary neutron star mergers). A neutron star is a compact astrophysical object whose interior serves as a cosmic laboratory for cold and dense QCD matter. Increasing astronomical observations—particularly those arising from the progress of gravitational wave analysis—will provide constraints on the extreme properties of QCD matter in this cold and dense regime, for which effective techniques for dealing with the associated inverse problem will be essential. Theoretically, first-principles lattice QCD calculations at vanishing and small baryon chemical potentials predict a smooth crossover transition from a dilute hadronic resonance gas to the deconfined QGP state. However, in the high-baryon density regime, direct lattice QCD simulations are currently hampered by the fermionic sign problem. On Earth, this new state of QGP matter can only be studied through heavy-ion collision (HIC) programs, where two heavy nuclei are accelerated and smashed to deposit the collision energy in the overlapping region for achieving extreme conditions, causing “heating and (or) compression” of the normal nuclear matter to be excited.

A significant challenge associated with HICs is that the collision of heavy nuclei is a highly dynamic, complex, and rapidly evolving process: although the deconfined QGP state may indeed be formed during the collision, it undergoes rapid expansion and cooling, and at some point its degrees of freedom are reconfined into color-neutral hadrons, which continue to interact and decay until the detector in the experiment receives its signals. The collision process is too short and too small to be resolved. Experimentally, we

have no direct access to the early potentially formed QGP fireball but only indirect measurements of the final emitted hadrons or their decay products. Furthermore, the theoretical description of the collision dynamics involves many uncertain physical factors that are not yet fully clear from theory or experimental comprehension. These uncertainties can interfere with final physical observables in the experiment. Thus, from the limited and contaminated (i.e., heavily influenced by many uncertain factors) measurements, a reliable extraction of the physics of the produced extreme QCD matter is non-trivial and challenging. This severely hampers the extraction of physical knowledge in the HIC programs.

As a modern computational paradigm, ML has become increasingly promising in recent years for applications at the forefront of HENP research. ML algorithms can be used to automatically identify patterns and correlations in data, allowing knowledge to be extracted from data computationally and automatically. It can thus help to extract meaningful information about the underlying physics or fundamental driving laws from the available data. In contrast to the traditional focus of ML, which is usually predictions based on pattern recognition from the collected data, the intersection of HENP and ML is concerned with the underlying patterns and causality for the purposes of uncertainty assessment and physical interpretation, which lead to discoveries. A collection of datasets from different areas of fundamental physics, including high-energy particle physics and nuclear physics, used for supervised ML studies was recently presented in Ref. [1].

For the purpose of physics identification, the intersection of HENP and ML goes beyond the mere application of existing learning algorithms to the dataset accessible in the physics problem. Paying special attention to the physical constraints or required fundamental laws or symmetries of the systems would increase the efficiency of ML in solving the specific physics problem. For example, when regressive or generative models are used to study quantum many-body systems or general quantum field theory (QFT), implementing the symmetries of the system can significantly reduce the amount of training data needed and improve the recognition performance [2]. ML has been applied in various studies at low- and intermediate-energy HICs [3–11]; a recent mini-review was presented in Ref. [12]. It has also been applied in hadron physics [13–15].

In addition, ML can be applied in the context of simulations, which play a key role in fundamental physics research as well as in a wide range of other scientific fields such as biology, chemistry, robotics, and climate modeling. In HENP, for both experimental and theoretical studies, simulation is an important tool, starting from the understanding of the fundamental interactions involved, e.g., in HIC dynamics and detector simulation, as well as in lattice QFT simulation. Simulations are used to model the behavior of

nuclear matter and its constituents and the interactions that occur between them, which are typically highly complex, with detailed use of many involved physical laws and equations or empirical phenomenological models. Simulations of HICs and the associated detectors in HENP consume large amounts of computational resources because of the high statistics and high resolutions. A collision dynamics simulation with extensive synthetic data is required to accurately interpret experimental measurements, which is enormously computationally and memory-intensive. ML can be used to improve the efficiency and descriptive power of these simulations to facilitate the physics discovery process. For example, researchers have proposed using ML to accelerate the simulation of hydrodynamics, to optimize the parameters involved in the model simulation, to make the model more robust to uncertainties, and to solve many-body problems directly by augmenting the conventional Monte Carlo simulation method.

In brief, ML is an effective tool that can be employed to address many challenges in HENP. It can assist in analyzing large amounts of data from HENP, linking nuclear experiments to physics theory exploration, optimizing simulations and calibrating models more efficiently, as well as developing new empirical and theoretical models. It is undeniable that ML technologies have the potential to make a significant impact, even transforming the field of HENP. Therefore, it is essential to acknowledge and recognize the importance of this new paradigm in advancing the field.

In the present review, focusing on HIC-related studies within HENP, we first provide a brief overview of the methodology in Sect. 2. Then, we discuss the applications of ML to HIC physics with regard to the following aspects: initial condition inference in Sect. 3, decoding bulk matter properties in Sect. 4, in-medium effects in Sect. 5, hard probe sector in Sect. 6, and searching for different observables in Sect. 7. We summarize our review in Sect. 8.

## 2 Methodology

### 2.1 Taxonomy of machine learning

ML can be classified in several ways. One way is to classify it by its function, i.e., into classification, regression, generation, and dimensionality reduction. The other way is to classify ML by the type of training data, i.e., into supervised learning, unsupervised learning, semi-supervised learning, self-supervised learning, active learning, and reinforcement learning. For example, supervised learning requires data to be labeled in such a way that the model can be trained to build a mapping between the input and the labels. Unsupervised learning does not need labeled data; it can learn patterns from data, assuming that the machine

makes self-consistent predictions on data that are perturbed or slightly augmented. Semi-supervised learning requires a small amount of labeled data along with a large amount of unlabeled data. Self-supervised learning works with specific data such as natural language or images that are sequential. It allows the machine to predict one part of the sequence from the other part. Active learning is a type of semi-supervised learning that employs two pools of data: a small pool of labeled data and a large pool of unlabeled data. The machine is trained on the labeled data and validated on the unlabeled data. The performance of the simply trained machine differs for different samples from the unlabeled data pool. For example, the machine may be uncertain on one sample, predicting that the label of the sample is A with 51% probability and B with 49% probability. This sample is assumed to be more difficult and more important for the trained machine than simple samples for which the machine's predictions are certain. For efficiency, this sample is labeled and moved from the unlabeled pool to the labeled pool for further training. Reinforcement learning uses data generated by interactions with the environment.

According to the previous description, the loss function for supervised learning in the regression task can be expressed as

$$l = \|y_{\text{pred}} - y_{\text{true}}\|, \quad (1)$$

where  $y_{\text{pred}} = f(x, \theta)$  is the function represented by ML models such as decision trees or deep neural networks (DNNs),  $x$  represents the input data,  $\theta$  represents all trainable model parameters, and  $y_{\text{true}}$  is the label of the input data  $x$ .  $\|\cdot\|$  usually represents the  $l_1$  norm, which gives the mean absolute error, or the  $l_2$  norm, which gives the root-mean-square error.

The cross-entropy loss is widely used for classification. It is defined as

$$l = - \sum_{k=1}^K p_k \log q_k, \quad (2)$$

where  $K$  represents the number of possible categories of the input data  $x$ ,  $p_k = y_{\text{true}}$  is the true label (probability), and  $q_k = f(x, \theta)$  represents the network prediction. This loss is inspired by the Kullback–Leibler (KL) divergence, which quantifies the difference between two distributions  $p$  and  $q$ :

$$\text{KL}(p||q) = \sum_{k=1}^K p_k \log \frac{p_k}{q_k} \quad (3)$$

$$= \sum_k p_k \log p_k - p_k \log q_k \quad (4)$$

$$= -H(p) + H(p, q), \quad (5)$$

where  $H(p)$  represents the entropy of the distribution  $p$  and the cross entropy  $H(p, q) = -\sum_k p_k \log q_k$  quantifies the average number of bits needed to encode the distribution  $p$  using the model  $q$ .

In binary classification, the cross entropy is reduced to

$$l = -\frac{1}{m} \sum_{i=1}^m [p_i \log q_i + (1 - p_i) \log(1 - q_i)], \tag{6}$$

where  $p_i$  is the true label of the  $i$ th sample, whose value is 0 or 1.  $q_i$  represents the network prediction obtained using the sigmoid activation function in the last layer to ensure  $0 < q_i < 1$ .  $m$  represents the number of samples in each mini-batch. If the true label is  $p_i = 0$ , only the 2nd part contributes to the loss function.

For multi-categorical classification, the loss function is the cross-entropy loss, with the activation function in the last layer replaced by the softmax function:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}. \tag{7}$$

For unsupervised learning, the loss function can generally be expressed as

$$l = ||\text{manipulate}_1(x) - \text{manipulate}_2(x)||, \tag{8}$$

where  $\text{manipulate}_{1,2}$  represents two manipulations on the same data. For example, in clustering tasks, the manipulations on  $x$  are to compute the total distance  $s$  of samples to multiple centers. In image classification tasks, the manipulations are to compute the network prediction over two different augmentations of the same image, e.g., cropping or rotation. This loss is also called self-consistent loss.

For semi-supervised learning, the loss function is the combination of the supervised loss and unsupervised loss:

$$l = l_{\text{supervised}} + l_{\text{unsupervised}}. \tag{9}$$

For self-supervised learning, a widely used loss function is the reconstruction loss. For example, in computer vision, the reconstruction loss is defined as the difference between the original image and the image reconstructed by a neural network from a masked image.

$$l = ||x - f((1 - M) \cdot x)|| \tag{10}$$

Here,  $x$  represents the original image,  $M$  represents the binary mask used to remove  $M = 0$  pixels from the image, and  $f$  represents a neural network used to reconstruct the image. The same method can be used to reconstruct natural language, by predicting the next sentence or missing words in a sentence. The pretrained network can be used in many downstream tasks, such as classification, regression, or generation.

In active learning, the loss function is essentially the same as that in supervised learning. The difference is that the trained network ranks samples from the unsupervised pool for annotation. Thus, the key is to rank the samples. There are two main methods for this. One is to rank the samples according to the entropy of the predictions made by the pretrained network:

$$s = -\sum_i p_i \log p_i, \tag{11}$$

where  $p_i$  represents the predicted probability that the sample is in class  $i$ . The other method is to rank the samples according to the diversity of the training dataset, by giving the highest rank to the sample that has the longest distance from the training data.

For reinforcement learning, the data are generated by subsequent interactions between the network policy and the environment. The network receives an observation  $o_t$  from the environment at time  $t$ , makes a decision, and performs an action  $a_t$  on the environment. The environment returns a new observation  $o_{t+1}$ , an immediate reward  $r_{t+1}$ , and a done signal. The data are thus  $\{o_t, a_t, o_{t+1}, r_{t+1}, \text{done}\}$  trajectories. The loss function of reinforcement learning is similar to that of supervised learning, with data  $o_t$  and true labels  $a_t, r_{t+1}$ .

### 2.2 Optimization

The goal of ML is to minimize the loss for the prediction of new data not used for training. In gradient-based models, this is achieved simply via stochastic gradient descent (SGD) and its variants:

$$\theta = \theta - \epsilon \frac{1}{m} \sum_{i=1}^m \frac{\partial l_i}{\partial \theta}, \tag{12}$$

where  $\theta$  represents all the trainable parameters of the ML model,  $\epsilon$  represents a small positive number called the learning rate, and  $m$  represents the size of the mini-batch. Updating  $\theta$  with the negative gradient  $-\epsilon \frac{1}{m} \sum_{i=1}^m \frac{\partial l_i}{\partial \theta}$  helps to gradually reduce the loss. This can be easily verified if there is only one trainable parameter  $\theta$  and the loss is  $l = \theta^2$ , whose negative gradient is  $-2\theta$ .

The possible values of  $\theta$  form a space called the parameter space. The initial value of  $\theta$  is usually a random number. Updating  $\theta$  using SGD is analogous to walking around the parameter space looking for the minimum value of the loss function. The loss function can be thought of as the potential surface whose negative gradients give the direction of acceleration  $\vec{a}$ . Thus, in simple SGD, the position  $\theta$  in the parameter space is updated using the acceleration. Consequently, native SGD has two major drawbacks. First, if the gradient is 0, the optimization stops immediately. Second, the network update is far faster in the direction where the gradient is large  $r$ . These

two drawbacks are partly solved using the momentum mechanism [16] and the adaptive learning rate [17].

In reinforcement learning, the goal is to maximize the accumulated rewards. The optimization method is stochastic gradient ascent. In the popular policy gradient method, the parameters of the policy network are updated as follows:

$$\theta = \theta + \epsilon G_t \nabla \ln \pi(a_t|o_t, \theta), \tag{13}$$

where  $G_t = \sum_{k=t+1}^T \gamma^{k-t-1} r_k$  is the return representing the accumulated rewards in the future with a discounting factor  $\gamma < 1$ .

### 2.3 Automatic differentiation

The number of trainable parameters in a DNN is large. To learn from the data, one must compute the negative gradients of loss with respect to each of the millions or trillions of model parameters  $-\frac{\partial l}{\partial \theta}$ . This is intractable using finite difference or analytic differentiation. Finite difference has both truncation and round-off errors that cannot be controlled. Analytical differentiation has exploding expressions for DNNs that are too complex to compute efficiently. In deep learning, the negative gradient is mainly computed using automatic differentiation (AD), which is computationally efficient meanwhile also has analytical precision.

AD has a forward mode and a backward mode. If the DNN is a  $R^1 \rightarrow R^n$  mapping, a forward pass gives derivatives of all output variables  $y_i$  with respect to the input variable  $x$ . In contrast, if the network is an  $R^n \rightarrow R^1$  mapping, each forward pass returns only the derivative of the output variable  $y$  on one of the input variables  $x_i$ . In the SGD algorithm, the backward mode is far more efficient because the mapping from  $\theta$  to the loss is an  $R^n \rightarrow R^1$  mapping. In the following, the forward AD is briefly explained.

In the forward mode, AD is implemented by introducing a dual number for each variable:

$$x \rightarrow x + \dot{x}\mathbf{d}, \tag{14}$$

$$y \rightarrow y + \dot{y}\mathbf{d}, \tag{15}$$

where  $x$  and  $y$  are two variables that require gradients, and  $\dot{x}$  and  $\dot{y}$  are the derivatives of  $x$  and  $y$ , respectively, with respect to some variable. As mentioned previously, with  $\dot{x} = 1, \dot{y} = 0$  gives  $\partial l/\partial x$  in one pass of the forward mode, and setting  $\dot{x} = 0, \dot{y} = 1$  gives  $\partial l/\partial y$  in another pass of the forward mode.  $\mathbf{d}$  is an infinitesimal symbol satisfying  $\mathbf{d}^2 = 0$ , which is analogous to the imaginary symbol  $\mathbf{I}^2 = -1$ . With this definition, the traditional output  $z$  of each operator is a dual number  $z + \dot{z}\mathbf{d}$  whose coefficient  $\dot{z}$  is the derivative of  $z$ , as follows:

$$x + \dot{x}\mathbf{d} + y + \dot{y}\mathbf{d} = (x + y) + (\dot{x} + \dot{y})\mathbf{d}, \tag{16}$$

$$x + \dot{x}\mathbf{d} - (y + \dot{y}\mathbf{d}) = (x - y) + (\dot{x} - \dot{y})\mathbf{d}, \tag{17}$$

$$(x + \dot{x}\mathbf{d}) * (y + \dot{y}\mathbf{d}) = x * y + (x\dot{y} + y\dot{x})\mathbf{d}, \tag{18}$$

$$(x + \dot{x}\mathbf{d})/(y + \dot{y}\mathbf{d}) = \frac{(x + \dot{x}\mathbf{d})(y - \dot{y}\mathbf{d})}{y^2 - \dot{y}^2\mathbf{d}^2} \tag{19}$$

$$= \frac{x}{y} + \frac{y\dot{x} - x\dot{y}}{y^2}\mathbf{d} \tag{20}$$

The calculations of dual numbers can easily be extended to polynomial functions:

$$P(x + \dot{x}\mathbf{d}) = P(x) + P'(x)\mathbf{d}. \tag{21}$$

Using a computer, more complex functions such as  $\sin(x)$ ,  $\log x$ , and  $e^x$  can be approximated by polynomial functions. In principle, AD works for these functions as well. In practice, these functions can be overloaded to produce outputs in the form of dual numbers, e.g.,  $\sin(x + \dot{x}\mathbf{d}) \rightarrow \sin x + \cos x\dot{x}\mathbf{d}$ .

Because of the universal approximation capability of DNNs and the efficient and accurate auto-diff, DNNs are widely used to represent solutions of ordinary differential equations (ODEs) and partial differential equations (PDEs) that require gradients. Thus, many physical problems are translated into optimization problems. This method is commonly referred to as physics-informed neural networks (PINNs). Compared with traditional numerical solutions, PINNs are mesh-free, work for very high dimensions, and are easy to implement—particularly for multi-scale and multi-physics problems.

### 2.4 Convolutional neural networks

Convolutional neural networks (CNNs) are distinguished from other neural networks by their superior performance for image, speech, and audio signal inputs. A naive CNN consists of three main types of layers, i.e., convolutional layers, pooling layers, and fully connected layers, as shown in Fig. 1.

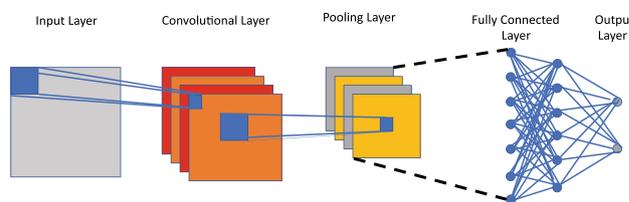


Fig. 1 (Color online) CNNs

The convolutional layer is the core building block of a CNN. The term convolution refers to the convolution operation between the input features and the filters (or kernels). In the mathematical view, a convolution operation is a special type of linear operation where two functions are multiplied to produce a third function that expresses how the shape of one function is modified by the other. In the ML view, the convolutional layer uses the filters to extract the features from the input data and combines the extracted features as the output. In a well-trained convolutional layer, a filter is only sensitive to one specific type of feature. Usually, there are many filters in a convolutional layer, to satisfy the complex input features. After the convolution operation, a rectified linear unit (ReLU) activation function is typically used, which introduces nonlinearity into the neural network.

After the convolutional layer, a pooling layer is applied to reduce the number of parameters, which is also known as downsampling. There are two main types of pooling: max pooling and average pooling. Max pooling selects the maximum value to be the output, and average pooling uses the average of the pixels covered by the pooling kernel. The fully connected layer is used to map the features extracted by the previous layers to the final output.

The convolutional layers can be stacked to make the neural network deeper. Earlier layers break down the complex features from the input data into individual simple features. As the features pass through the subsequent convolutional layers, the filters begin to capture larger elements or shapes. Owing to its ability to extract complex features, the CNN architecture became a foundation of modern computer vision.

However, when neural networks are deep, the vanishing gradient problem is severe. To overcome this problem in CNN architectures, many complex neural networks have been developed, such as AlexNet, VGGNet, InceptionNet, GoogLeNet, and ResNet.

## 2.5 Recurrent neural networks

Recurrent neural networks (RNNs) are distinguished from other neural networks by their superior performance for sequence or time-series data.

Figure 2 shows the structure of a basic RNN, where  $U$  denotes the weights for the connection of the input layer to the hidden layer,  $V$  denotes the weights for the connection of the hidden layer to the hidden layer, and  $W$  denotes the weights for the connection of the hidden layer to the output layer. Using self-connection with weights  $V$ , the RNN takes information from previous inputs to influence the current input and output. This feature, which is often referred to as “memory”, makes the RNN good at processing sequential data. The loss function  $\mathcal{L}$  of all timesteps is defined according to the loss at each timestep, as follows:

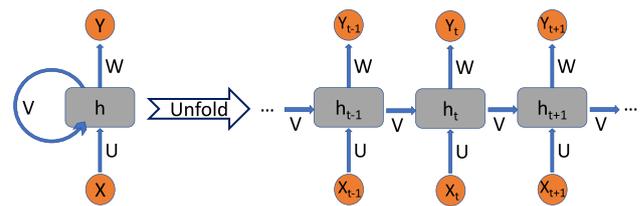


Fig. 2 (Color online) RNNs

$$\mathcal{L}(\hat{Y}, Y) = \sum_{t=1}^T \mathcal{L}(\hat{Y}^t, Y^t). \quad (22)$$

The RNN uses the backpropagation through time (BPTT) algorithm to determine the gradients. The error is back-propagated from the last timestep to the first timestep. At timestep  $T$ , the derivative of the loss  $\mathcal{L}$  with respect to the weight matrix  $W$  is expressed as follows:

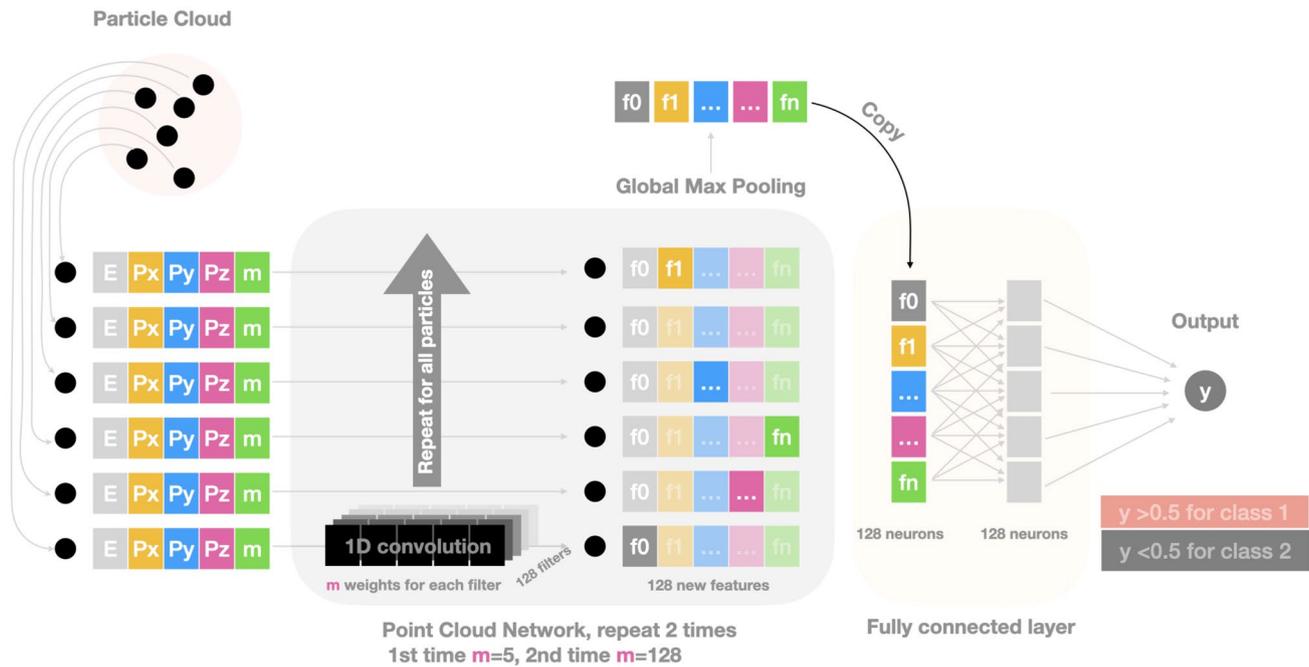
$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(t)}}{\partial W}. \quad (23)$$

RNNs also suffer from the problems of gradient vanishing and exploding. To deal with the gradient problems, variant networks have been developed, such as long short-term memory (LSTM) networks and gated recurrent units (GRUs).

## 2.6 Point cloud network

The final-state particles from HICs form a point cloud in the momentum space. The data must be manipulated to use CNNs and RNNs, as these networks were originally designed for images and natural language. For example, to use a CNN, density estimation (histogram) is typically used to convert the particle cloud into images. However, this does not work well for a few particles in a three-dimensional (3D) space, because the particles are dilute and the resolution is poor. To use an RNN, the particle cloud must be sorted to one dimension, which can only keep the local information in one dimension. The point cloud network is designed to preserve the permutation symmetry of a set of particles.

Figure 3 shows a simple demonstration of a point cloud network. The input to the network is a set of particles in the momentum space, including their 4-momenta, mass, and other quantum numbers. A fully connected neural network or multilayer perceptron (MLP) is applied to a particle to transform its  $m$  input features into 128 features in the high-dimensional latent space. The MLP is shared by all the particles in the cloud and is also called a 1DCNN. This step preserves the permutation symmetry of all the particles. Then, global max pooling (GMP) or global average pooling (GAP) is applied to these latent



**Fig. 3** (Color online) Simple example of a point cloud network

features of the particles to extract the global information of the particle cloud. The GMP and GAP extract the boundaries of the input particle cloud in the high-dimensional latent space, which learn the multi-particle correlation for the final decision. This extracted global information (128 features) is fed to another MLP for the final decision. The output neuron has a value in the range (0, 1) and uses 0.5 as the decision boundary.

The network shown in Fig. 3 is used to classify nuclear phase transitions [18]. Some point cloud networks apply a Euclidean rotation to the point cloud to preserve rotational symmetry, i.e., the network should make self-consistent prediction if the point cloud is rotated globally [19]. Other variants use k-nearest neighbors in the spatial or momentum space to extract the high-dimensional latent features of each particle, for keeping more local correlation. The k-nearest neighbors of each particle can be calculated in the feature space to capture the long-range multiple particle correlation, because particles that are close in the feature space may be far apart in the spatial or momentum space. This technique is called dynamical edge convolution and was used to search for self-similarity between particles in the momentum space, which is associated with critical phenomena that may occur in HICs [20]. The dynamical edge convolutional neural network is a type of message-passing neural network that is also called a graph neural network.

## 2.7 Generative modeling

In unsupervised learning, generative modeling is a class of techniques related to probability distribution learning. With regard to tasks, ML can generally be categorized into discriminative modeling and generative modeling. From probabilistic perspectives, discriminative modeling, such as pattern recognition, aims at learning a conditional probability  $p(y|x)$ , which can be used to make predictions for a given input object ( $x$ ) its associated properties or class identities ( $y$ ), while the goal of generative modeling is to capture the joint distribution  $p(x, y)$ , from which one can generate new data points following the same statistics as the training set. Generative modeling has achieved considerable success in numerous applications, including image synthesis, inpainting, super-resolution, text-to-image translation, speech generation, and chat robots. Many of the generative models were developed with profound influence from and on physics. Generative modeling also has numerous direct applications in science, e.g., computational fluid simulation, drug molecule design, anomaly detection, many-body physics, and lattice field configuration generation for QCD.

The central purpose of generative modeling is to sample data ( $\tilde{x}$ ) from the same distribution of the training set  $p_d(x)$ . Most of the generative modes construct parametric (explicit or implicit) models  $p_\theta(x)$  to approach the desired data distribution. From information theory, the KL divergence (Eq.(3)),

which measures the dissimilarity between the model and data distributions, provides an objective for this task. Per Jensen inequality, the KL divergence is non-negative and is zero only when the two distributions match exactly. The minimization of the KL divergence under given observational data for the system with the collected training set, i.e.,  $\mathcal{D} = \{x\}$ , is equivalent to the minimization of the negative log-likelihood (NLL):

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log p_{\theta}(x); \quad (24)$$

thus, maximum likelihood estimation (MLE) is performed.

In the following, we briefly review several representative and popular deep generative models, including the variational autoencoder (VAE), generative adversarial networks (GANs), autoregressive modeling, and normalizing flows (NF).

VAE [21], introduces a latent variable  $z$  to facilitate the generation process; thus, it constructs a trainable conditional probability  $p_{\theta}(x|z)$  (called the decoder or generator, usually modeled by a neural network). For generation convenience, the latent variable is assumed to follow an easy-to-sample prior distribution  $p(z)$ , such as the multivariate Gaussian distribution. However, the introduction of the latent variable makes the data generation distribution (thus the likelihood) intractable, because the required marginalization is  $p_{\theta}(x) = \int p_{\theta}(x|z)p(z)dz$ . Thus, the posterior distribution for the latent variable is intractable as well, because  $p(z|x) = p_{\theta}(x|z)p(z)/p_{\theta}(x)$ . The VAE employs a variational inference approach to approximately perform MLE on the training data. Specifically, an encoder model  $q_{\phi}(z|x)$  (also a neural network) is introduced to approach the real posterior  $p(z|x)$ , and the KL divergence  $\mathcal{D}_{\text{KL}}(q_{\phi}(z|x)||p(z|x))$  provides the training objective, which is derived as a variational lower bound (also known as an evidence lower bound (ELBO)), which is the cornerstone of the VAE) of the likelihood:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x|z) + \log p(z) \\ &\quad - \log q_{\phi}(z|x)] \leq \log p_{\theta}(x), \end{aligned} \quad (25)$$

$$\theta, \phi = \arg \max_{\theta, \phi} \mathcal{L}. \quad (26)$$

The *generative adversarial network*, i.e., GAN, as another latent variable generative model, is developed to train the generator through an adversarial strategy. Intuitively, the GAN framework constructs two nonlinear differentiable functions (both represented by adaptive neural networks with dimensionality accordingly set). The first one—called the generator  $G(z)$ —maps latent variable  $z$  to the target data manifold  $\tilde{x} = G(z)$ , which gives an implicit synthesized data distribution  $p_G(x)$  when the latent variable follows a prior latent distribution  $p(z)$ , e.g., multivariate uniform or Gaussian, and the goal is training the generator to drive  $p_G(x)$

approaching the target distribution  $p_{\text{true}}(x)$ . The other one—called the discriminator  $D(x)$ —maps the data manifold to a single scalar representing the fake-vs.-true distinguishing result of the discriminator for the input data. For a vanilla GAN, the discriminator is designed as a binary classifier; i.e., for real and generated data, it is trained to output  $D(x) = 1$  and  $D(\tilde{x}) = 0$ , respectively. The generator and discriminator are trained alternately to improve their abilities in competing against each other; this can be achieved by mimicking a two-player min-max game. Thus, the discriminator is trained to better distinguish the real data from the generated data, while the generator is trained to trick the discriminator into classifying generated data as “real” data.

It was proven mathematically that the adversarial training of a GAN is equivalent to minimizing the Jensen–Shannon divergence:

$$\text{JS}(p_{\text{real}}||p_G) = \frac{1}{2}(\text{KL}(p_{\text{real}}||p_{\text{mix}}) + \text{KL}(p_G||p_{\text{mix}})), \quad (27)$$

with  $p_{\text{mix}} = (p_{\text{real}} + p_G)/2$ . Thus, the GAN is an implicit MLE-based generative model. The optimally trained GAN converges in the Nash equilibrium state, where the generator excels in synthesizing samples that the discriminator cannot differentiate from the real data; thus, the generator-induced distribution indeed captures the real data distribution after training. This technique has been utilized in various scientific contexts, e.g., in condensed-matter physics [22, 23], particle physics [24, 25], cosmology [26, 27], and QFT study with lattice simulation [28, 29].

### 2.7.1 Autoregressive model

There are also explicit MLE-based generative models, which are closely related to statistical physics. Among them, the simplest is the autoregressive model [30], which invokes the probability chain rule to decompose the full probability into products of a series of conditionals:

$$p_{\theta}(x) = \prod_i^N p_{\theta}(x_i|x_1, x_2, \dots, x_{i-1}), \quad (28)$$

which are used as the generative model distribution to approach the desired data distribution. Specifically, neural networks can be used to parameterize each of the conditional components in the above equation. Then, these neural networks can be viewed as a single general neural network (having a fully connected or CNN or RNN architecture) with a masked weight parameter matrix (e.g., triangular with matrix for the simple fully connected case) considering the autoregressive properties specified by Eq. 28. The use of a convolutional layer or recurrent layer—called PixelCNN [31] or PexelRNN [32], respectively—for treating structured systems in autoregressive modeling can further

account for the spatial or temporal translational invariance of the system. It has achieved state-of-the-art performance in speech synthesis with autoregressive networks such as WaveNet [33]. With the above autoregressive representation as a parametric generative model, MLE can be explicitly performed to optimize the  $p_\theta(x)$  for approaching the targeted data distribution  $p_{\text{real}}(x)$ , which as derived is minimizing the forward KL divergence  $\text{KL}(p_{\text{real}}||p_\theta)$ . This idea is also applied in many-body physics for the study of statistical mechanics and general continuous systems [34].

### 2.7.2 Normalizing flow

The NF [35–37] combines the latent variable model and explicit MLE. It introduces bijective affine transformations to map a simple latent space variable  $z$  to the complex data manifold sample  $x = g(z)$ . The bijectivity requires the transformation to have the same dimensionality in the input and output. This allows for the usage of the change of variable theorem to estimate the likelihood explicitly:

$$p_\theta(x) = p(z) \left| \det \left( \frac{\partial z}{\partial x} \right) \right|, \tag{29}$$

with the determinant of the Jacobian for the (inverse) transformation needed. Then, after the MLE training, the parameterized transformation serves as a generator for new sample generation  $x = g(z)$ . To simplify the evaluation of the needed Jacobian determinant in Eq. 29, special network structures are adopted, e.g., those holding the triangular Jacobian matrix, as used in Real NVP. Such flow-based generative models have been implemented in lattice QFT studies [38–40] and have proven to be useful indicators for QCD study in the past few years. Recently, such a flow-based model was generalized into the Fourier frequency space and used in generating Feynman paths for quantum physics [41].

## 2.8 Principal component analysis

In ML, principal component analysis (PCA) is a statistical technique that involves transforming a set of correlated variables into independent variables through orthogonal transformations. The principal components, which are associated with the obtained main eigenvectors (or non-negligible singular values), reveal the most representative configurations of the data. As one of the unsupervised learning techniques, PCA implements singular value decomposition (SVD) on a real matrix [42]:

$$\mathbf{M} = \mathbf{X}\mathbf{\Sigma}\mathbf{Z} = \mathbf{V}\mathbf{Z}, \tag{30}$$

where  $\mathbf{M}$  is a matrix of size  $N \times m$ ;  $\mathbf{X}$  and  $\mathbf{Z}$  are two orthogonal matrices of size  $N \times N$  and  $m \times m$ , respectively; and  $\mathbf{\Sigma}$

is a diagonal matrix with the singular values arranged in descending order. Then, the  $i_{\text{th}}$  row of the matrix  $\mathbf{M}^{(i)}$  can be expressed as

$$\begin{aligned} \mathbf{M}^{(i)} &= \sum_{j=1}^m x_j^{(i)} \sigma_j z_j = \sum_{j=1}^m \tilde{v}_j^{(i)} z_j \\ &\approx \sum_{j=1}^k \tilde{v}_j^{(i)} z_j \quad (i) = 1, \dots, N \end{aligned} \tag{31}$$

where  $\tilde{v}_j^{(i)}$  is the corresponding coefficient of  $z_j$  for the  $i_{\text{th}}$  row. In the last step, there is a cut on the indices, i.e.,  $k$ , because PCA focuses on the most important components. Owing to its effectiveness for data mining, PCA has been widely used in various areas of physics research. For recent progress in HICs, please see Sect.7.

## 3 Initial condition

In the traditional view, the nuclear structure manifests its significance only at low energy, because the high-energy nucleus–nucleus collisions are violent processes in which the whole nucleus is disassembled. However, recent findings have indicated that the initial nuclear structure information is very important for understanding the final observables in high-energy HICs. One of the examples is collective flows, e.g., elliptic flows and triangular flows, in which the initial participant shape and nucleon density distribution, as well as their initial state fluctuations, are relevant. In particular, the collision geometry, neutron skin, deformation, and  $\alpha$ -clustering structure significantly affect the final observable. A mini-review can be found in a chapter in the handbook of nuclear physics authored by Ma and Zhang [43]. ML is a powerful tool for discriminating such initial structure information. In this section, we discuss such applications.

### 3.1 Impact parameter estimation

The impact parameter  $b$  describes the distance between the centers of the two colliding nuclei in the classical view, which is a crucial quantity determining the initial geometry of a collision. In experiments, the impact parameter is not directly measurable and is usually estimated from the multiplicity of final-state particles in track detectors or the energy deposited in calorimeters. ML approaches are proposed to determine the impact parameters from the final-state particles and exhibit better performance than conventional methods. Ref. [44] proposed the use of a DNN and CNN to reconstruct the impact parameters from the energy spectra of final-state charged hadrons of HICs at  $\sqrt{s_{\text{NN}}} = 7.7$  to

200 GeV, which were simulated with a multiphase transport (AMPT) model. Both the DNN and CNN can reconstruct the impact parameters with an MAE of approximately 0.4 fm. When the input feature is from a larger pseudorapidity window, the CNN has a higher prediction accuracy than the DNN. Ref. [45] reported the performance of a CNN and Light Gradient Boosting Machine (LightGBM) in reconstructing the impact parameter from HICs at  $\sqrt{s_{NN}} = 0.2$  to 1 GeV, which were simulated with the Ultra-relativistic Quantum Molecular Dynamics (UrQMD) model. The input features are constructed from the proton spectra for transverse momentum and rapidity. The average difference between the true impact parameter and the estimated one can be less than 0.1 fm. LightGBM has better performance than the CNN.

A model-independent Bayesian inference method for reconstructing the impact parameter distributions was proposed in Ref. [46]. The impact parameter distributions are inferred from model-independent data. This method is based on Bayes' theorem:

$$P(b | \mathbf{X}) = P(b)P(\mathbf{X} | b)/P(\mathbf{X}), \quad (32)$$

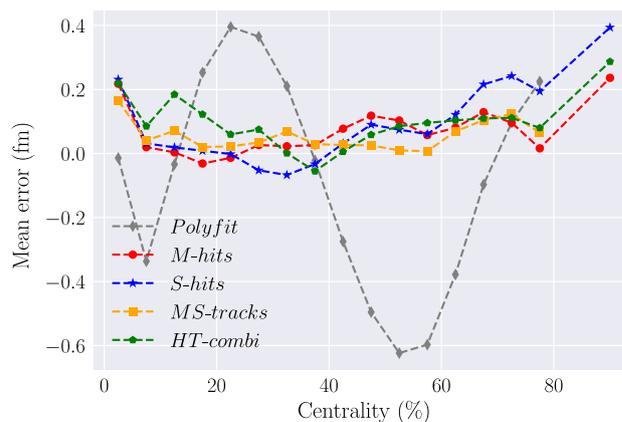
where  $P(\mathbf{X})$  represents the probability of the observable that can be measured in the experiment, and  $P(\mathbf{X} | b)$  represents the probability density distribution of  $\mathbf{X}$  for a given impact parameter  $b$ . Fluctuation is taken into account by assuming  $P(\mathbf{X} | b)$  to be a Gaussian or gamma distribution, which can be determined by fitting the data with the formula  $P(\mathbf{X}) = \int P(\mathbf{X} | b)P(b)db$ .  $P(\mathbf{X})$  can be multidimensional. In Ref. [46] two observables were used:  $\mathbf{X} = \{M, p_t^{\text{tot}}\}$ , where  $M$  represents the multiplicity of the charged particles and  $p_t^{\text{tot}}$  represents the total transverse momentum of the light particles.

### 3.2 End-to-end centrality estimation for CBM

The compressed baryonic matter (CBM) detector is currently under construction for the Facility for Antiproton and Ion Research (FAIR) at Gesellschaft für Schwerionenforschung (GSI), which will study the properties of strongly compressed nuclear matter via HICs with beam energies ranging from 2 to 10 AGeV. A characteristic of the CBM experiment is its very high event rate and trigger rate, which will produce a large amount of raw data per second in real-time and pose a challenge for online event characterization and storage. To address the online event characterization, it is essential to be able to work on the direct output of the detector, which has an inherent point cloud structure—a collection of points as an unordered list with particles or tracks' attributes recorded. One important property of the point cloud is that they as a whole should be invariant under permutation. The PointNet structure [47] was specially developed to respect this order invariance. Accordingly, for

HICs, PointNet-based models can perform real-time physics analysis on the detector output directly.

References [19, 48] proposed the use of PointNet-based models for event-by-event impact parameter determination for the CBM experiment using the direct output from the detector, where the trained model serves as an end-to-end centrality estimator. The supervised learning strategy is used for this regression task, where the training data are prepared from UrQMD followed by CBMRoot detector simulation to obtain the detector output, which are hits or tracks of the particles. A PointNet-based model is constructed and trained to capture the inverse mapping between the detector output and the impact parameter information. It was shown that PointNet-based models can perform accurate event-by-event impact parameter determination using hits of charged particles in different detector planes and/or the tracks reconstructed from these hits. With regard to both precision and accuracy, these models outperformed a baseline model using charged track multiplicity as the input inside a polynomial fit. While the baseline model had a similar resolution (relative precision) to the PointNet-based model in the semi-central collision region, it had a lower accuracy and more fluctuations in the accuracy for impact parameters ranging from 3 to 16 fm, as indicated by the mean of the prediction error for the impact parameter. This trend was more evident for a realistic event distribution (i.e.,  $\sim bdb$ ), as shown in Fig. 4 for the mean prediction error. Considering the natural parallelizability and high speed, the PointNet-based model paves the way for real-time end-to-end event characterization for HIC studies.

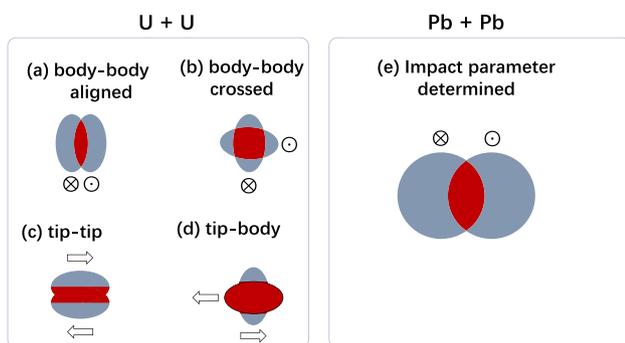


**Fig. 4** (Color online) Taken from Ref. [48]. Mean error in predictions as a function of centrality. Dataset *Test2* is used, in which peripheral events are more likely to occur than other centralities. The track multiplicity is used for the centrality binning. The points at 90% centrality are results from events with no tracks reconstructed. Therefore, the *Polyfit* and *MS-Tracks* models do not have a data point at 90% centrality

### 3.3 Nuclear deformation estimation

The momentum distribution of final-state hadrons is sensitive to nuclear shape deformation. For example, owing to the different collision geometry, the elliptic flow as a function of charge multiplicity differs significantly between Pb+Pb and U+U collisions. As shown in Fig. 5, the  $^{208}\text{Pb}$  is a doubly magic nucleus with an almost perfectly spherical shape, whose collision patterns depend only on the impact parameter  $b$ . In contrast, the shape of  $^{238}\text{U}$  is similar to a watermelon, and the corresponding collision patterns are far more complex than those of Pb+Pb collisions. For example, the U+U collisions have body–body aligned, body–body crossed, tip–tip, and tip–body collisions. Different collision patterns correspond to different charge multiplicities and elliptic flows. Both the fully overlapped body–body aligned and central tip–tip collisions correspond to most central collisions with high charge multiplicity, but their elliptic flows differ significantly. This type of difference leads to a far larger variance in the elliptic flow for most central U+U collisions, compared with high-multiplicity Pb+Pb collisions. In principle, the complex collision patterns lead to many differences in the elliptic flow compared with the charged multiplicity diagram. Deep learning can be used to identify these differences and predict the nuclear shape deformation parameters using these patterns.

It was demonstrated that by using nuclei with different deformation parameters  $\beta_2$  and  $\beta_4$ , high-energy HICs can be simulated using the TRENTo Monte Carlo model to obtain the event-by-event total initial entropy (which is proportional to the final charged multiplicity) and the corresponding geometric eccentricity (which is approximately proportional to the elliptic flow). A deep residual neural network was trained to predict  $\beta_2$  and  $\beta_4$  using the two-dimensional (2D) images of total entropy vs. eccentricity [49]. The network accurately predicted the absolute values of  $\beta_2$  and  $\beta_4$  but failed to predict their signs using the information provided. Using the class activation map (CAM) method to map the last convolutional



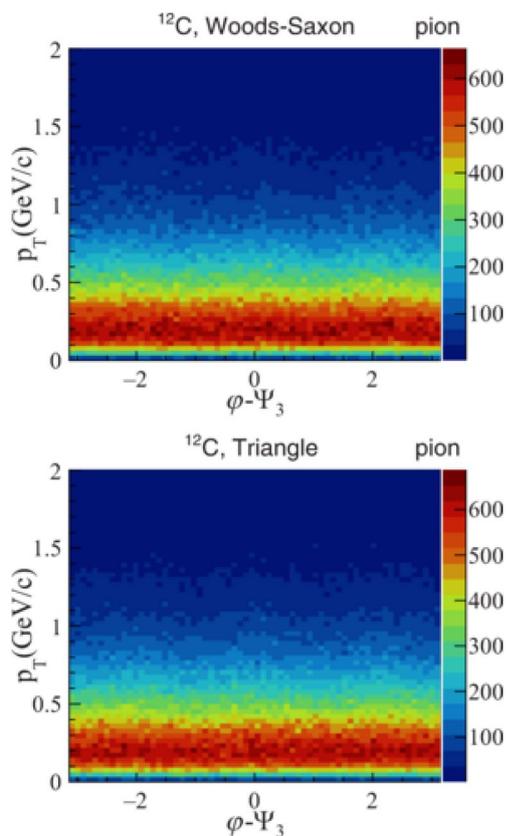
**Fig. 5** (Color online) Collision geometries for Pb+Pb and U+U collisions

layer onto the input image, the authors found two regions in the image that are important for decision-making. One is the most central collision region, which is the most sensitive region to the variance of the elliptical flow.

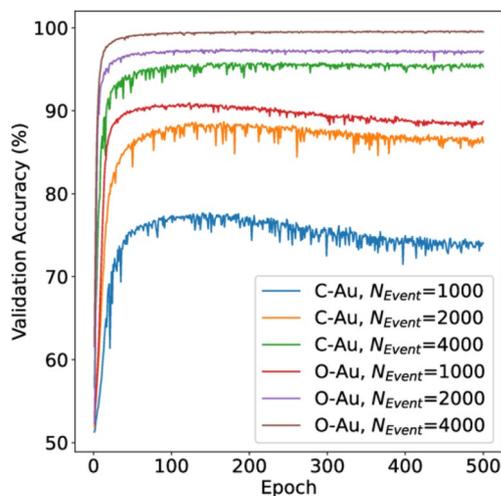
Recently, Bayesian inference with a Gaussian process (GP) emulator was used for reconstructing the nuclear structure including deformation parameters based on HIC measurements [50]. As a first-step exploratory study, the collision observables (charged multiplicities  $N_{\text{ch}}$ , elliptic flow  $v_2$ , triangular flow  $v_3$ , and mean transverse momentum  $\langle p_T \rangle$ ) were estimated via Monte Carlo Glauber model calculation (total energy  $E$ , elliptic eccentricity  $\epsilon_2$ , triangular eccentricity  $\epsilon_3$ , and energy density  $d_{\perp}$ ), can reasonably estimate the ratio of observables in isobaric collision systems owing to the cancelation of dynamics' uncertainties [51]. Under this setup, nuclear structure reconstruction based on both single collision system and contrast isobaric collision system measurements were discussed. For single collision systems, it was found that the Woods–Saxon parameters of nuclei can be precisely inferred from final-state observables estimated with  $(P, \epsilon_2, \epsilon_3, d_{\perp})$ . For isobar collision systems, the simultaneous inference on the two set of nuclear structures fails with only the ratio of these final observables, while the further provision of the single collision system's multiplicity distribution allows high-precision nuclear structure reconstruction. Additionally, the ratio of radial flow was found to be redundant in the presence of the ratio of elliptic flow and vice versa.

### 3.4 $\alpha$ -clustering structure

The clustering structure is an exotic phenomenon in nuclei, and it usually occurs in light nuclei [52]. In nuclear collisions between light clustering nuclei and heavy ions, the clustering structure can make the final-state particles anisotropically distributed [43, 53, 54]. It is crucial to extract the quantitative information about the clustering from the final observables. In the  $^{12}\text{C}/^{16}\text{O} + ^{197}\text{Au}$  collisions at relativistic energies, an ML method was used to obtain evidence of the cluster structures from the azimuthal angle and transverse momentum distributions of charged pions [55]. In this study, a Bayesian convolutional neural network (BCNN) was used. In addition to the input layer and output layer, there were hidden layers, consisting of four convolutional layers and three fully connected layers. The parameters of the three fully connected layers were sampled from distributions learned via Bayesian inference. A 2D histogram of azimuthal angle vs. transverse momentum was used as the input. Considering the detection efficiency in the experiments, charged pions with rapidity ranging from  $-1$  to  $1$  and transverse momentum ranging from  $0$  to  $2$  GeV/ $c$  were selected. The dataset consisted of  $1.6 \times 10^6$  histograms with  $64 \times 64$  bins (pixels), with different labels to indicate different configurations.



**Fig. 6** (Color online) Taken from Ref. [55]. Two-dimensional azimuthal angle vs transverse momentum distributions of charged pions for non-clustered (Up) and clustered (Down)  $^{12}\text{C}$  from an AMPT-generated  $^{12}\text{C}+^{197}\text{Au}$  collision event at  $\sqrt{s_{\text{NN}}} = 200$  GeV



**Fig. 7** (Color online) Taken from Ref. [55]. Validation accuracy during the training process for colliding systems  $^{12}\text{C}/^{16}\text{O} + ^{197}\text{Au}$  with  $N_{\text{Event}} = 1000, 2000,$  and  $4000$

The typical spectra of 4000 merged events are shown in Fig. 6. Even with merging, the samples of different configurations are barely distinguishable to the naked eye. The number of merged events is denoted as  $N_{\text{Event}}$ , which is taken to be 1000, 2000, and 4000.

The learning curves are shown in Fig. 7. As more events were merged, the event-by-event fluctuations were reduced, and the network was able to learn the features of the final state for predicting the initial configuration. For  $^{12}\text{C}$  with  $N_{\text{Event}} = 4000$  and  $^{16}\text{O}$  with  $N_{\text{Event}} = 2000$ , the validation accuracy reached 95% and 97%, respectively, and for  $^{16}\text{O}$  with  $N_{\text{Event}} = 4000$ , it reached 99%.

For the clustering phenomenon, it is extremely difficult to extract signals from the final particles, because fluctuations play such an important role in relativistic HICs. By averaging over multiple events, the BCNN model can learn the features with good performance.

### 3.5 Neutron skin estimation

The distribution of neutrons is important in determining the thickness of the neutron skin, the symmetry energy of the nucleus, the QCD equation of state (EoS) of dense nuclear matter, and astrophysical observables such as the mass–radius relationship of neutron stars and the gravitational wave emitted during neutron star mergers. However, extracting the distribution of neutrons inside the nucleus is extremely difficult. The distribution of neutrons inside the nucleus differs from the distribution of protons. The proton distribution is far easier to measure than the neutron distribution because the former is equivalent to the charge distribution, whereas the latter is associated with the weak charge distribution. The neutron skin, which is the difference between the root-mean-square radii of neutrons and protons, can be used to determine the neutron (weak charge) distribution in the nucleus. PREX2 measured the parity-violating asymmetry by scattering longitudinally polarized electrons on  $^{208}\text{Pb}$  to obtain a neutron skin thickness of approximately  $R_n - R_p = 0.283 \pm 0.071$  fm [56]. The neutron skin is used as a constraint in the calculation of the positive and negative correlations between the symmetry energy and the slope parameter at the saturation density. With this constraint, the Bayesian analysis achieves a compromise between the “conflicting” data that lead to the famous “PREXII puzzle” and the “soft Tin puzzle” [57, 58].

There have been many attempts to determine the neutron skin thickness and the symmetry energy at low energy [59], e.g., by investigating the charge-exchange spin–dipole excitation [60], the supernova neutrinos [61], nuclear fragmentation reactions [62], and parity-violating electron scattering [56, 63].

For high-energy HICs, it was proposed that the isobar ratios of the charge multiplicities of the mean transverse

momentum and the net charge multiplicities between  $^{96}_{44}\text{Ru} + ^{96}_{44}\text{Ru}$  and  $^{96}_{40}\text{Zr} + ^{96}_{40}\text{Zr}$  can be used to precisely determine the nucleon skin and the symmetry energy [64]. The authors claimed that the high-energy isobar collisions can significantly improve the result of the traditional low-energy method. In another paper, the yields of spectator protons and neutrons at the forward velocity of ultra-central collisions were proposed to be good probes of the neutron skin—sensitive to the neutron skin of  $^{208}\text{Pb}$  but insensitive to other parameters during the collision [65]. A more accurate method is to measure the free spectator neutron yield ratios between  $^{96}_{44}\text{Ru} + ^{96}_{44}\text{Ru}$  and  $^{96}_{40}\text{Zr} + ^{96}_{40}\text{Zr}$  in ultra-central collisions [66].

A large amount of data have already been collected from high-energy HICs. There may be a data-driven way to reuse these data to determine the neutron distribution and neutron skin thickness. It has been tested in [67] that nucleons sampled from nuclei with different neutron skin types can be classified with reasonable accuracy using deep CNNs and point cloud networks. However, once the nucleus is involved in HICs, it is almost impossible to distinguish the neutron skin types of the colliding nucleus using the momentum distribution of the final-state hadrons. For this task, the signal is weak in minimum bias collisions, and DNNs fail to solve the difficult inverse problem. A new ML method is needed to search for weak signals in data with large statistical fluctuations.

## 4 Bulk matter

### 4.1 Shear and bulk viscosities

The shear and bulk viscosities are important properties that significantly affect the dynamical expansion of QGP and the momentum distribution of final-state hadrons, as indicated by relativistic fluid dynamics simulations [68–71]. In solving the inverse problem of HICs, it was found that the effects of viscosity are entangled with the initial thermalization time, the EoS of QGP, and the phase transition between QGP and HRG. Thus, determining the shear and bulk viscosities of hot nuclear matter is a notoriously difficult problem. Regarding the nucleonic degree of freedom, the shear viscosity has attracted considerable attention because it is related to the nuclear EoS, phase change, and strong interaction [72–75]. A similar feature to the QGP viscosity has been demonstrated for the behavior  $\eta/s(T)$ . Bayesian analysis plays an important role in determining the temperature dependence of the ratio of shear viscosity over the entropy density ratio  $\eta/s(T)$  as well as the bulk viscosity over the entropy density ratio  $\zeta/s(T)$  [76–78].

Suppose that all the parameters in the theoretical model of HICs form a set  $\{\theta\}$  and all the experimental data from RHIC and LHC form another set  $\{D\}$ . Then, the posterior distribution of the model parameters is given by

$$P(\theta_i|D) = \frac{P(D|\theta_i)P(\theta_i)}{P(D)} = \frac{P(D|\theta_i)P(\theta_i)}{\sum_j P(D|\theta_j)P(\theta_j)}, \quad (33)$$

where  $P(D|\theta_i)$  represents the likelihood between the experimental data  $D$  and model output using parameter combinations  $\theta_i$ ;  $P(\theta_i)$  represents the a priori distribution of  $\theta_i$ , which may be a belief based on past experience or physical considerations; and the denominator  $P(D) = \sum_j P(D|\theta_j)P(\theta_j)$  is a normalization factor called evidence. Computing  $P(D)$  is too expensive because it requires the theoretical model to traverse the entire parameter space. Fortunately, in Bayesian analysis, the normalization factor is not needed, because the Markov chain Monte Carlo (MCMC) method can sample from the following un-normalized distributions:

$$P(\theta_i|D) \propto P(D|\theta_i)P(\theta_i). \quad (34)$$

The final output of the Bayesian analysis is a large number of different combinations of model parameters sampled from the above un-normalized posterior distribution function. Performing a density estimation for each parameter, e.g., the slope of  $\eta/s$  over  $T_c$ , gives a distribution (or histogram) of the slope parameter. The location of the maximum value in this distribution corresponds to the MAP estimate. Additionally, the distribution has a variance that corresponds to the uncertainty in the slope parameter, which comes from the experimental data, the prior distribution, and the likelihood function. Thus, it is clear that the extracted model parameters are well constrained when their posterior distribution has a narrow peak.

To estimate the temperature dependence of the shear and bulk viscosities, two parameterized functions based on physical a priori are required. In a Nature Physics paper [77], the shear and bulk viscosities were parameterized as follows:

$$(\eta/s)(T) = (\eta/s)_{\min} + (\eta/s)_{\text{slope}}(T - T_c) \left(\frac{T}{T_c}\right)^{(\eta/s)_{\text{crv}}}, \quad (35)$$

$$(\zeta/s)(T) = \frac{(\zeta/s)_{\max}}{1 + \left(\frac{T - (\zeta/s)_{T_{\text{peak}}}}{(\zeta/s)_{\text{width}}}\right)^2}, \quad (36)$$

where  $(\eta/s)_{\min}$  and  $(\zeta/s)_{\max}$  represent the minimum and maximum shear and bulk viscosity values to be determined, respectively,  $T_c = 154$  MeV is the QCD transition temperature representing the location of the minimum in  $\eta/s(T)$ , and  $(\zeta/s)_{T_{\text{peak}}}$  denotes the location of the maximum bulk viscosity to be determined. Other parameters to be determined are the

slope  $(\eta/s)_{\text{slope}}$  and the curvature of the shear viscosity  $(\eta/s)_{\text{crv}}$  and the width of the bulk viscosity peak  $(\zeta/s)_{\text{width}}$ .

Without considering other parameters, these six parameters form a six-dimensional parameter space. The above Bayes formulae are used to traverse this space, with the trajectories forming a set of parameter combinations. This is equivalent to importance sampling using the posterior distribution of the six parameters. Density estimation indicates that the distribution of  $(\eta/s)_{\text{min}}$  is approximately normal, whose mean and variance give a quantitative estimate of  $(\eta/s)_{\text{min}} = 0.085^{+0.026}_{-0.025}$ . An anti-correlation is observed between  $(\zeta/s)_{\text{max}}$  and  $(\zeta/s)_{\text{width}}$ , indicating that it is the integral of  $(\zeta/s)(T)$  that matters, not its specific form. The analysis also indicates that the experimental data used can not constrain the parameters  $(\eta/s)_{\text{crv}}$  and  $(\zeta/s)_{T_{\text{peak}}}$ , as there are no obvious peaks in the posterior distributions of these two parameters.

### 4.2 Crossover or first-order phase transition

In general, as mentioned in the Introduction, the challenge faced by high-energy nuclear collision studies can essentially be viewed as an inverse problem. Assuming that all related physical factors (e.g., initial condition/fluctuations, QGP bulk properties, transport coefficients, freeze-out parameter, hadronic interactions) are given, well-established theoretical models (e.g., relativistic viscous hydrodynamics with hadronic transport simulation) can be adopted to simulate the HIC process to give their final-state observables, and such a forward process is well understood. However, given instead only limited measurements of the final state of HICs, it is unclear how to disentangle those different influencing physical factors for decoding the corresponding early time dynamics. For high-energy HICs, there are two strategies for solving this inverse problem using statistical methods and ML: one is Bayesian inference with the task of parameter estimation for calibrating the chosen model (e.g., in Ref. [79]), and the other is supervised ML for directly capturing the inverse mapping from the final state to the corresponding physics of interest.

Reference [80] proposed the use of a deep CNN to capture the direct inverse mapping from the final-state information to the types of QCD transition happened in early

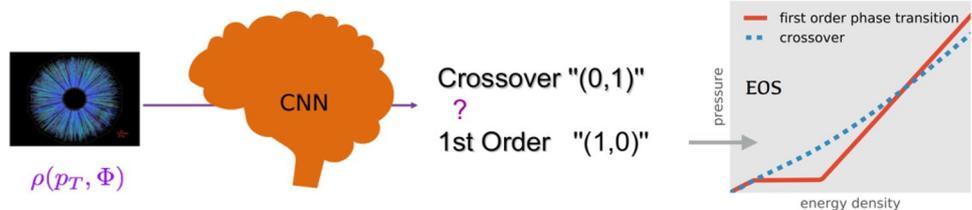
time. This is inspired by the success of image recognition in computer vision. Although the inverse mapping may be very implicit, DNNs can be used to decode it and represent it in the sense of Big Data in a supervised manner. The required training data can be prepared through well-established model simulation for HICs, e.g., using the state-of-the-art 3+1-dimensional viscous hydrodynamics [81–84], where diversity can be introduced by varying different physical factors (i.e., parameters in the simulation). As an exploratory study, a binary classification task was targeted, where the Deep CNN was trained to identify the QCD transition type embedded within the collision dynamics as crossover or first-order solely according to the final pion spectra  $\rho(p_T, \phi)$ , as shown in Fig. 8. The EoS of the hot and dense matter is a crucial ingredient in the hydrodynamic simulations. Embedded in it is the nature of the QCD transition (first-order or crossover), which can significantly affect the hydrodynamic evolution according to the shape of the pressure gradient. As the input to the deep CNN, the final charged pion’s spectra at mid-rapidity are obtained using the Cooper–Frye formula in each hydrodynamic simulation:

$$\rho(p_T, \phi) = \frac{dN_i}{dY p_T dp_T d\phi} = g_i \int_{\sigma} p^{\mu} d\sigma_{\mu} f_i(p \cdot u), \quad (37)$$

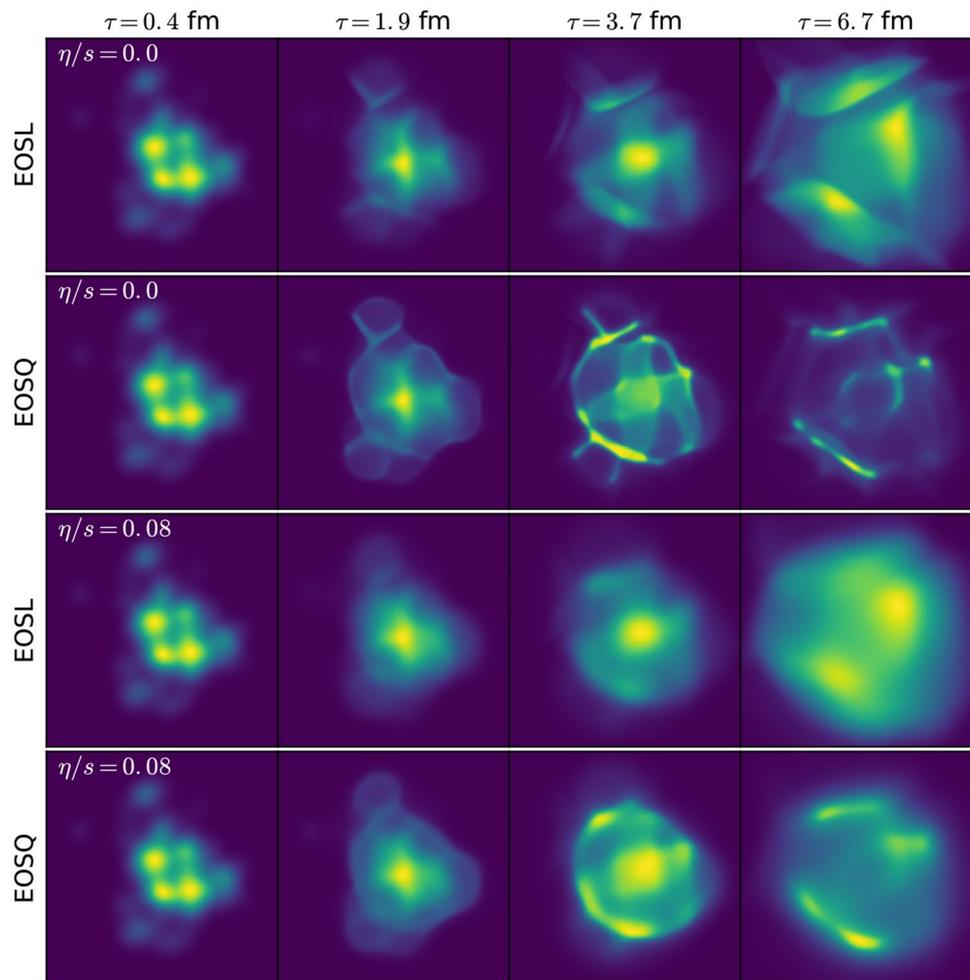
where  $N_i$  represents the particle number density,  $Y$  represents the rapidity,  $g_i$  represents the degeneracy,  $d\sigma_{\mu}$  represents the freeze-out hypersurface element, and  $f_i$  represents the thermal distribution. The training dataset of  $\rho(p_T, \phi)$  was generated from the event-by-event hydrodynamic package CLVisc [81] with fluctuating AMPT initial conditions, with which supervised learning using the CNN is performed for binary classification in identifying the QCD transition types.

Figure 9 shows the space–time evolution histories of QGP expansion starting from the same initial condition model with different fluctuations, in relativistic hydrodynamic simulations using CLVisc. For EOSQ with a first-order phase transition, the pressure gradient is zero in the mixed phase. Multiple ridge structures are formed with a first-order phase transition in the EoS because the expansion of QGP is mainly driven by the pressure gradient and the acceleration is 0 in the mixed phase. However, the expansion histories are significantly different when the shear viscosity is not

**Fig. 8** (Color online) Schematic of QCD transition classification with HIC final particle spectra



**Fig. 9** Evolution history of QGP simulated using the relativistic hydrodynamic model CLVisc, starting from the same initial condition with four different parameter combinations. From top to bottom, each row presents four snapshots taken at different times, using different combinations of the EoS and shear viscosity over the entropy density ratio. Here, EOSL represents the lattice QCD EoS with a crossover transition between QGP and hadron resonance gas, and EOSQ represents an EoS with a first-order phase transition between QGP and hadron resonance gas



0. Different evolution histories lead to different final-state particle spectra in the momentum space.

To verify the robustness of the trained deep CNN in this QCD EoS recognition task, the test set was simulated from a different hydrodynamics package or with different initial fluctuating conditions (IP-Glasma or MC-Glauber) and different  $\eta/s$  parameters. The conventional observables, such as the elliptic flows  $v_2$  and integrated particle spectrum, were shown to be insufficient for distinguishing the two QCD transition classes for these test set, whereas the trained deep CNN achieved an average classification accuracy of 95%, indicating that it was robust against contamination from factors such as the initial fluctuations and shear viscosity. For comparison, the best classification accuracy among traditional ML algorithms such as decision tree, random forest, support vector machine, and gradient boosting was approximately 80%. The good performance of the trained deep CNN indicates that the imprint of the early time transition dynamics is not fully washed out by the collision evolution and is still embedded in the final-state information. Additionally, the inverse mapping from final-state observables to

the QCD transition information can be well captured by the deep CNN from the supervised training strategy, providing a discriminative and traceable encoder for the dynamical information of QCD transition. Thus, the constructed deep CNN functions as an “EoS-meter” to efficiently bridge the HIC experiments to QCD bulk matter physics. This study paved a path to the success of experimental research on the QCD EoS and the search for the critical endpoint of the QCD phase diagram. In the study, the afterburner hadronic cascade effects were not considered; thus, the conclusion regarding the direct inverse mapping was drawn from the viewpoint of pure hydrodynamic evolution.

Later, this strategy was deepened in a series of studies for more realistic scenarios, e.g., to take into account the afterburner hadronic cascade by incorporating UrQMD following the hydrodynamics evolution [85, 86]; to consider non-equilibrium dynamics of the phase transition’s influence, e.g., spinodal decomposition [18, 87] or Langevin dynamics [88]; to include more realistic experimental detector effects through detector simulation with hits or tracks as the input [48, 89]; to perform unsupervised outlier

detection for HICs [90]; and to determine the nuclear symmetry energy [91]. Specifically, in Ref. [89] it was shown that by using just the detector output directly, PointNet models can be employed to classify collision events simulated by an EoS associated with a first-order phase transition and those simulated by an EoS with a crossover transition. The PointNet models take the reconstructed tracks from the CBM detector (simulated with CBMRoot) followed by the hybrid UrQMD events. They achieved a binary classification accuracy of approximately 96% when trained on collision events for impact parameters ranging from 0 to 7 fm. When the model training set was shrunk to the mid-central region with  $b = 0 \sim 3$  fm, the model accuracy increased to approximately 99%. A combination of training sets from both peripheral and mid-central collisions resulted in a classifier being able to identify the phase transition type across different centralities, while not compromising the accuracy for the central region.

### 4.3 Active learning for QCD EoS

First-principles calculations using lattice QCD provide the EoS of hot nuclear matter at high temperatures and zero baryon chemical potential. Because of the fermionic sign problem, lattice QCD fails to compute the nuclear EoS at finite  $\mu_B$  at present. Using Taylor expansion, it is possible to obtain the nuclear EoS at a small  $\mu_B$  that is close to zero, approximately. The BEST collaboration formulated a nuclear EoS with a critical endpoint by mapping the 3D Ising model with the Taylor expansion result. However, the model contains four free parameters whose values determine the size and location of the critical endpoint. Some combinations of these parameters lead to an unphysical, e.g., acausal or unstable, EoS.

Supervised learning can help to map unphysical regions of parameter combinations. However, labeling is computationally expensive in this task. For thermodynamic stability, one must check the positivity of the energy density, pressure, entropy density, baryon density, second-order baryon susceptibility  $\chi_2^B$ , and heat capacity  $(\partial S/\partial T)_{n_B}$ , as well as the causality condition:

$$0 \leq c_s^2 \leq 1, \quad (38)$$

where  $c_s$  represents the speed of sound in hot nuclear matter.

Active learning was used to find the most informative parameter combinations before labeling them [92]. In active learning, the network is first trained using a small amount of labeled data. Then, the trained network is employed to make predictions on all samples from a large unsupervised pool. If the network is uncertain about one parameter combination, e.g., it predicts that this group of parameter combinations will lead to an EoS that is unphysical with probability 51%,

this sample lives on the decision boundary and should be informative and important for the network. Labeling this sample will improve the performance of the network more than labeling easy samples. The newly labeled sample is moved out of the pool and will be used in supervised learning later.

### 4.4 Accelerated relativistic hydrodynamic simulation via deep learning

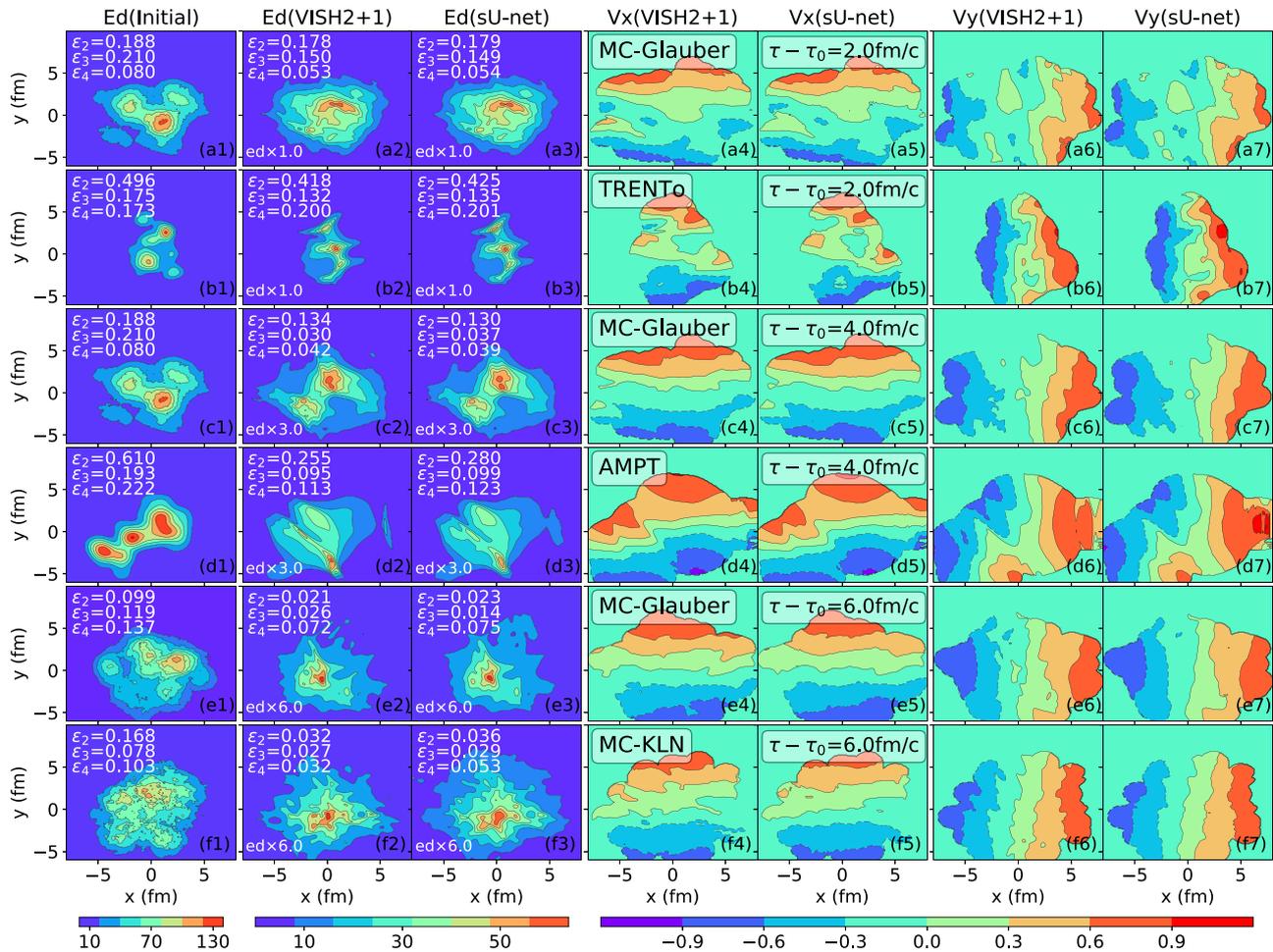
Relativistic hydrodynamics is a powerful tool for simulating the QGP expansion and studying the flow observables in relativistic HICs at the RHIC and LHC energies [95–100]. For ideal hydrodynamics with zero net charge densities, it solves the transport equations of the energy momentum tensor:

$$\partial_\mu T^{\mu\nu} = 0, \quad (39)$$

where  $T^{\mu\nu} = (e + p)u^\mu u^\nu - pg^{\mu\nu}$ ,  $e$  represents the energy density,  $p$  represents the pressure, and  $u^\mu$  represents the four-velocity. In traditional hydrodynamic simulations, these transport equations are numerically solved with an algorithm such as SHASTA or LCPFCT that transforms the initial conditions into final-state profiles through nonlinear evolutions [97, 100–102].

Recently [93, 94], a DNN called stacked U-net (sU-net) was designed and trained to learn the initial- and final-state mapping from the nonlinear hydrodynamic evolution. The constructed sU-net has an encoder–decoder architecture, which contains four U-net blocks with residual connections between them. For each U-net block, there are three convolutional and deconvolutional layers with Leaky ReLU and softplus activation functions employed for the inner and output layers, respectively. By concatenating the feature maps along the channel dimension, the output of the first two convolutional layers is fed to the last two deconvolution layers. For details, please refer to [93, 94].

The training and test data (the profiles of the initial and final energy momentum tensor  $T^{\tau\tau}$ ,  $T^{\tau x}$ ,  $T^{\tau y}$ ) were generated from VISH2+1 hydrodynamics [103, 104] with zero viscosity, zero net baryon density, and longitudinal boost invariance. In more detail, sU-net was trained with 10,000 initial and final profiles from VISH2+1 with MC-Glauber initial conditions [105, 106], and then its prediction accuracy was tested using the profiles of four different types of initial conditions: MC-Glauber [105, 106], MC-KLN [106, 107], AMPT [81, 108, 109], and TRENTo [110]. Figure 10 presents the final energy density and flow velocity predicted by sU-net, together with a comparison with the hydrodynamic results. As shown, the trained sU-net captured the magnitudes and structures of both the energy density and the flow velocity. In particular, panels (b), (d), and (f) show that the network, which was trained with datasets generated with



**Fig. 10** (Color online) Energy density and flow velocity profiles predicted by sU-net and calculated from VISH2+1 for six test initial profiles of MC-Glauber, MC-KLN, AMPT, and TRENTo. Taken from Refs. [93, 94]

MC-Glauber initial conditions, was also capable of predicting the final profiles of other types of initial conditions. In Refs. [93, 94], the eccentricity coefficients, which indicate the deformation and inhomogeneity of a large number of the energy density profiles, were calculated, and the predictions from sU-net almost overlapped with the results from VISH2+1.

Compared with the 10~20 min simulation time of VISH2+1 on a traditional CPU, sU-net took several seconds to directly generate the final profiles for different types of initial conditions on one P40 GPU, which significantly accelerated the traditional hydrodynamic simulations. However, the sU-net model designed and trained in Refs. [93, 94] mainly focuses on mimicking the 2+1-dimensional hydrodynamic evolution with a fixed evolution time. For more realistic implementation, it is important to explore the possibilities of mapping the initial profiles to the final profiles of the particles emitted on the freeze-out surface of the relativistic HICs.

## 5 In-medium effects

### 5.1 Spectral function reconstruction

Accessing real-time properties of QCD (or a many-body system in general) remains a notoriously difficult problem, because the non-perturbative computations, such as lattice field simulations or functional methods, usually operate in Euclidean space-time (after a Wick rotation  $t \rightarrow it \equiv \tau$ ) and thus can only provide Euclidean correlators (i.e., in imaginary time). Thus, the analytic continuation of these discrete noisy data is often ill-posed. Quantitatively understanding the real-time dynamics determined by the Minkowski correlator is important and interesting, e.g., for understanding scattering processes, transport, or non-equilibrium phenomena that occur in HICs. The Minkowski correlator is usually accessed from the Euclidean correlator via spectral reconstruction.

The associated ill-posed problem can be cast as a Fredholm equation of the first kind:

$$g(t) = \int_a^b K(t, s)\rho(s)ds, \quad (40)$$

with the goal of retrieving the function  $\rho(s)$  given the kernel function  $K(t, s)$  but limited information about  $g(t)$ . It has been well shown that the required inverse transform becomes ill-conditioned if only a finite set of data points with non-vanishing uncertainty are available for  $g(t)$ . In the context of QFT, one can simply approach this problem via the Källén–Lehmann spectral representation of the correlators, taking the kernel function to be

$$K(t, s) = s(s^2 + t^2)^{-1}\pi^{-1}, \quad (41)$$

where the  $\rho(s)$  functions involved are usually called spectral functions. The task of reconstructing the spectral function from the correlator measurements (from the lattice calculation) needs to be regularized to make sense of the inverse problem involved. Over the past few decades, many different regularization techniques have been explored for this ill-conditioned inverse problem, such as Tikhonov regularization, maximum entropy methods, and Bayesian inference techniques.

Recently, deep learning-based strategies have also been explored to tackle spectral reconstruction, which can be mainly categorized into two schemes: data-driven supervised learning approaches and unsupervised learning-based approaches. The first application of domain-knowledge-free deep-learning methods to this ill-conditioned spectral reconstruction (also called analytic continuation) was reported in Ref. [111] in the context of general quantum many-body physics. The results indicated the good performance of DNNs with supervised training in the cases of a Mott–Hubbard insulator and a metallic spectrum. In particular, a CNN was found to achieve better reconstruction than a fully connected network, with performance superior to that of the MEM—one of the most widely used conventional methods. In Ref. [112], the authors adopted a similar strategy but also introduced PCA to reduce the dimensionality of the QMC-simulated imaginary time correlation function of the position operator for a harmonic oscillator linearly coupled to an ideal heat bath.

The authors of Ref. [113] also adopted a data-driven perspective. They adopted a strategy similar to spectral function reconstruction in the QFT context and considered the Källén–Lehmann spectral representation as the accessible propagator, i.e.,  $G(p) = \int_0^\infty \frac{d\omega}{\pi} \frac{\omega\rho(\omega)}{\omega^2 + p^2}$ , which takes the kernel in the Fredholm equation as the Källén–Lehmann kernel. For the dummy spectral functions, the superposition of Breit–Wigner peaks was used, according to the perturbative one-loop QFT-derived

parameterization  $\rho^{\text{BW}}(\omega) = 4\Gamma\omega/((M^2 + \Gamma^2 - \omega^2)^2 + 4\Gamma^2\omega^2)$ . Two types of DNNs have been studied, both with a noisy propagator as the input, but with different outputs: one estimates the parameters (e.g.,  $\Gamma_i$  and  $M_i$  for the collection of Breit–Wigner peaks) of the spectral function (denoted as PaNet), and the other attempts to directly reconstruct the discretized data points of the spectral function (denoted as PoNet).

As another type of nonparametric representation, GPs were used in the reconstruction of the 2+1 flavor QCD ghost and gluon spectral function in Ref. [115]. In general, the GP can define a probability distribution over families of functions, which is typically characterized by the chosen kernel function. In Ref. [115] the GP was assumed to describe the spectral function:

$$\rho(\omega) \sim \mathcal{GP}(\mu(\omega), C(\omega, \omega')), \quad (42)$$

where the mean function  $\mu(\omega)$  is often set to zero, and the covariance  $C(\omega, \omega')$  is determined by the kernel function used, for which a common standard choice is the radial basis function (RBF) kernel

$$C(\omega, \omega') = \sigma_C e^{-\frac{(\omega-\omega')^2}{2l^2}}, \quad (43)$$

with tunable hyperparameters  $\sigma_C$  for the overall magnitude and  $l$  for the length scale. The prior represented by this GP can be plugged into the Bayesian inference procedure with lattice data for the ghost dressing function and gluon propagator for evaluating the likelihood. In Ref. [115], the lattice data were specifically extended. The ghost dressing function was extended to the deep infrared range, and the low-frequency behavior was constrained by spectral DSE results [116]. The gluon dressing function was extended to the ultraviolet range with previous fRG computation results [117]. This reduced the variance in the solution space and enhanced the stability compared with the inference without such extensions. It was shown that while approximately fulfilling the Oehme–Zimmermann superconvergence (OZS) condition for gluons, the reconstruction with GP regression in this work accurately reproduced the lattice data within the uncertainties with deviations for a gluon propagator stronger in some regions than those for the ghost dressing function. For the spectral function, the reconstruction exhibited a similar peak structure to a previous fRG reconstruction of the Yang–Mills propagator [117].

In Refs. [114, 118, 119], the authors developed an unsupervised approach based on DNN representation for the spectral function together with automatic differentiation (AD) to reconstruct the spectral function, which does not need training data preparation for supervision (a similar DNN-based inverse problem solving strategy within the AD framework was used for reconstructing the neutron-star

EoS from astrophysical observables [120, 121] and inferring the parton distribution function of pions in lattice QCD studies [122]). The introduced DNN representation can preserve the smoothness of the spectral function automatically, helping to regularize the degeneracy issue in this inverse problem. This is because, as analyzed in Ref. [119], the degeneracy is related to the null modes of the investigated kernel function, which usually induce oscillation for the reconstructed spectral function. Specifically, the DNN-represented spectral function, i.e.,  $\vec{\rho} = [\rho_1, \rho_2, \dots, \rho_{N_\omega}]$ , can be converted into the propagator under the discretization scheme as  $D(p) = \sum_i^{N_\omega} K(p, \omega_i) \rho_i \Delta\omega$ . Then, the loss function over the propagator relative to lattice data, i.e.,  $\mathcal{L} = \sum_i^{N_p} (D_i - D(p_i))^2 / \sigma_i$ , can be evaluated and provide guidance for tuning over the DNN-represented spectral function. Taking gradient-based algorithms, the derivative of the loss with respect to network parameters can be derived as

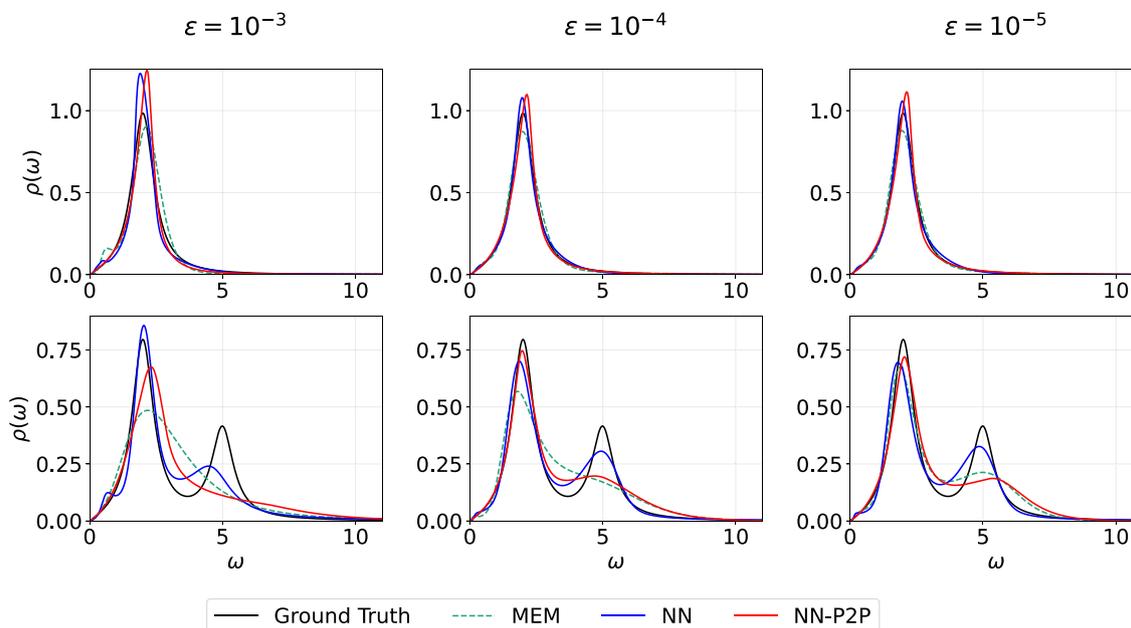
$$\nabla_{\theta} \mathcal{L} = \sum_{j,k} K(p_j, \omega_k) \frac{\partial \mathcal{L}}{\partial D(p_j)} \nabla_{\theta} \rho_k. \tag{44}$$

$\nabla_{\theta} \rho_k$  is computed easily under standard backward propagation for the network.

For the DNN representation of the spectral function, two different schemes were investigated in this work: one uses the multiple outputs of an L-layer neural network to represent in list format the spectral function (denoted as NN), and the other directly uses a feedforward neural network for parameterization (denoted as NN-P2P) of the spectral

function as a function of frequency, i.e.,  $\rho(\omega)$ . For the training, the Adam optimizer is adopted, and the  $L_2$  regularization is set in the warm-up beginning stage under an annealing strategy until the regularization strength value is sufficiently small (set as  $< 10^{-8}$  in the calculation). This can relax the regularization to obtain hyperparameter-independent inference results. For the direct NN list representation, a quenched implementation of smoothness condition  $\lambda_s \sum_{i=1}^{N_\omega} (\rho_i - \rho_{i-1})^2$  is also performed with  $\lambda_s$  reduced from  $10^{-2}$  to 0. This unsupervised spectral reconstruction method was validated with regard to the uniqueness of the solutions both analytically and numerically [119]. As shown in Fig. 11, for superposed Breit–Wigner peaks, this method outperformed the traditional MEM method—particularly for multi-peak spectra with large amounts of measurement noise.

In addition to Gaussian-like and Lorentzian-like spectral reconstruction tests, the newly devised framework presented in Refs. [114, 118] was validated through two physics-motivated tests. One was for non-positive definite spectral reconstruction, which is beyond the scope of classical MEM applicability but is often encountered for spectral functions related to confinement phenomenon of, e.g., gluons and ghosts, or thermal excitations with long-range correlation in strongly coupled systems. The other one was for the hadron spectral function encoded in the temperature-dependent thermal correlator with lattice QCD noise-level noises. For both of these physical cases, the proposed DNN and AD-based method with NN representation consistently works



**Fig. 11** (Color online) Spectral functions reconstructed from MEM, NN, and NN-P2P under different amounts of Gaussian noises added to the propagator data with  $N_p = 25$ , and  $N_\omega = 500$ . Taken from Ref. [114]

well, whereas traditional MEM based methods lose the peak information or fail to resolve the non-positiveness.

The spectral function can also be reconstructed from finite correlation data by implementing the radial basis function network (RBFN), which is an MLP model based on the RBF [127, 128]. The RBFN has been widely used in feature extraction, classification, regression, etc. [129–132]. In Ref. [123], the spectral function  $\rho(\omega)$  was approximately described by a linear combination of RBFs:

$$\rho(\omega) = \sum_{j=1}^N w_j \phi(\omega - m_j), \tag{45}$$

where  $\phi$  represents the active RBF with an adjustable weight  $w_j$  and an adjustable center  $m_j$ , which can take a Gaussian form  $\phi(r) = e^{-\frac{r^2}{2a^2}}$  or an MQ form  $\phi(r) = (r^2 + a^2)^{\frac{1}{2}}$ . Here,  $a$  is the shape parameter, which is adjustable and essential for the regularization. Then, the inverse mapping problem of constructing the spectral function is transformed into calculating the linear weights of the RBF, which allows smooth and continuous reconstruction.

For calculating these parameters in Eq. (45), in Ref. [123], a neural network called the RBFN was constructed, which is a three-layer feedforward neural network with the active RBFs in the hidden layer. After discretization of the spectral function, Eq.(45) is converted to matrix form:  $[\rho] = [\Phi][W]$ . Then, the correlation functions in the Euclidean space with the integral spectral representation  $G(\tau, T) = \int_0^\infty \frac{d\omega}{2\pi} \rho(\omega, T) K(\omega, \tau, T)$  are converted to matrix form:

$$G_i = \sum_{j=1}^M \sum_{k=1}^N K_{ij} \Phi_{jk} w_k \equiv \sum_{k=1}^M \tilde{K}_{ik} w_k, \quad i = 1 \dots \hat{N}, \tag{46}$$

where  $\tilde{K}$  is a  $\hat{N} \times M$  matrix associated with the integration kernel, and  $\hat{N}$  represents the number of data points for the correlation function  $G_i$ . The spectral function  $\rho(\omega_i)$  has been discretized into  $N$  parts with  $m_i = \omega_i$ ,  $i = 1 \dots N$ , and  $M$  is set to  $M = N = 500$ . To obtain  $w_j$ , one can use the truncated singular value decomposition (TSVD) method or a DNN [123]. Compared with other ML approaches based on supervised learning [113, 133, 134], this method allows faster training and is free from the overfitting problem.

Figure 12 shows a comparison of the spectral functions reconstructed using RBFN, TSVD, Tikhonov, and MEM, using the correlation data generated by a mock SPF. The mock SPF was obtained by mixing two Breit–Wigner distributions:  $\rho_{\text{Mock}}(\omega) = \rho_{\text{BW}}(A_1, \Gamma_1, M_1, \omega) + \rho_{\text{BW}}(A_2, \Gamma_2, M_2, \omega)$  with  $\rho_{\text{BW}}(A_i, \Gamma_i, M_i, \omega) = \frac{4A_i \Gamma_i \omega}{(M_i^2 + \Gamma_i^2 - \omega^2)^2 + 4\Gamma_i^2 \omega^2}$ . The parameters for the mock SPF in Fig. 12 were set to  $A_1 = 0.8, M_1 = 2, \Gamma_1 = 0.5; A_2 = 1, M_2 = 5, \Gamma_2 = 0.5$ . Here, 30 discrete correlation data were generated using the Euclidean

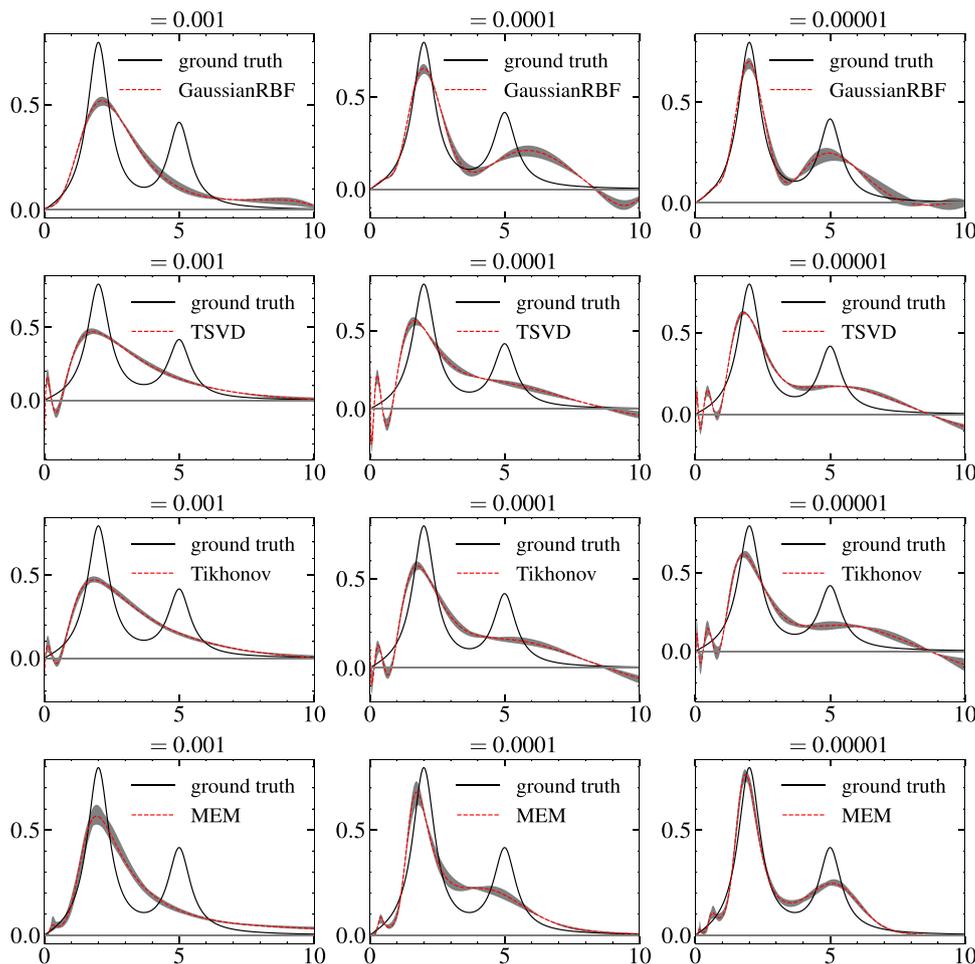
correlation functions of the mock SPF, with noise added, i.e.,  $G_{\text{noise}}(\tau_i) = G(\tau_i) + \text{noise}$ .

Compared with the results of traditional methods, the RBFN provided a better description of the spectral functions—particularly for the low-frequency part. It almost reproduced the first peak of the mock SPF using the correlation data with a small amount of noise  $\epsilon = 0.00001$ . In contrast, Tikhonov, TSVD, and MEM exhibited oscillation behavior at a low frequency. For such a task of extracting the transport coefficients from the Kubo relation, an improved reconstruction of the spectral functions at a low frequency is important. Although the RBFN failed to reconstruct the second peak of the mock SPF, it was the only method that reduced the oscillation at the low frequency, among the methods tested. In Ref. [123], the Gaussian and MQ RBFs used in the network were compared, and it was found that the Gaussian RBF provided better construction of the SPF, including the location and the width of the peak. Additionally, with mock data generated from the spectral function of the energy momentum tensor, it was demonstrated that the RBFN method allows precise and stable extraction of the transport coefficients.

### 5.2 In-medium heavy quark potential

As an important probe for the properties of the created QGP in HICs, heavy quarkonium (the bound state of a heavy quark and its anti-quark) has been intensively measured in experiments and analyzed in theoretical studies [135, 136], wherein the investigation and calculation require an understanding of the in-medium heavy quark interaction. The heavy quarkonium provides a calibrated QCD force, because in vacuum the simple Cornell potential can well reproduce the spectroscopy of heavy quarkonium, and when we put the bound state into the QCD medium, the color screening effects naturally occur and weaken the interactions between the heavy quarks, beyond which a non-vanishing imaginary part manifested as thermal width is argued to appear according to both one-loop hard thermal loop (HTL) perturbative QCD calculations [137, 138] and recent effective field theory (EFT) studies, e.g., those on PNRQCD [139, 140]. However, a non-perturbative treatment similar to that of lattice QCD is necessary because it is difficult to obtain a satisfactory description of the strong interaction dictated in-medium heavy quarkonium solely from perturbative calculations. These EFT studies suggested that a potential-based picture can provide a good approximation of the quarkonium, under which the Schrödinger equation can be employed to study the spectroscopy of the bound state. Recent lattice QCD studies involved quantification of the in-medium spectrum–mass shift and thermal widths of bottomonium ( $b\bar{b}$ ) up to 3 S and 2P states in QGP [125], where it was found cannot be reproduced by the one-loop HTL-motivated functional

**Fig. 12** (Color online) Constructed spectral functions obtained from RBFN, TSVD, Tikhonov regularization, and MEM, using the correlation data generated by the mock SPF acquired by mixing two Breit–Wigner distributions. From left to right, different Gaussian noises are added to the correlation data with  $\epsilon = 0.001, 0.0001, \text{ and } 0.00001$ . Taken from Ref. [123]

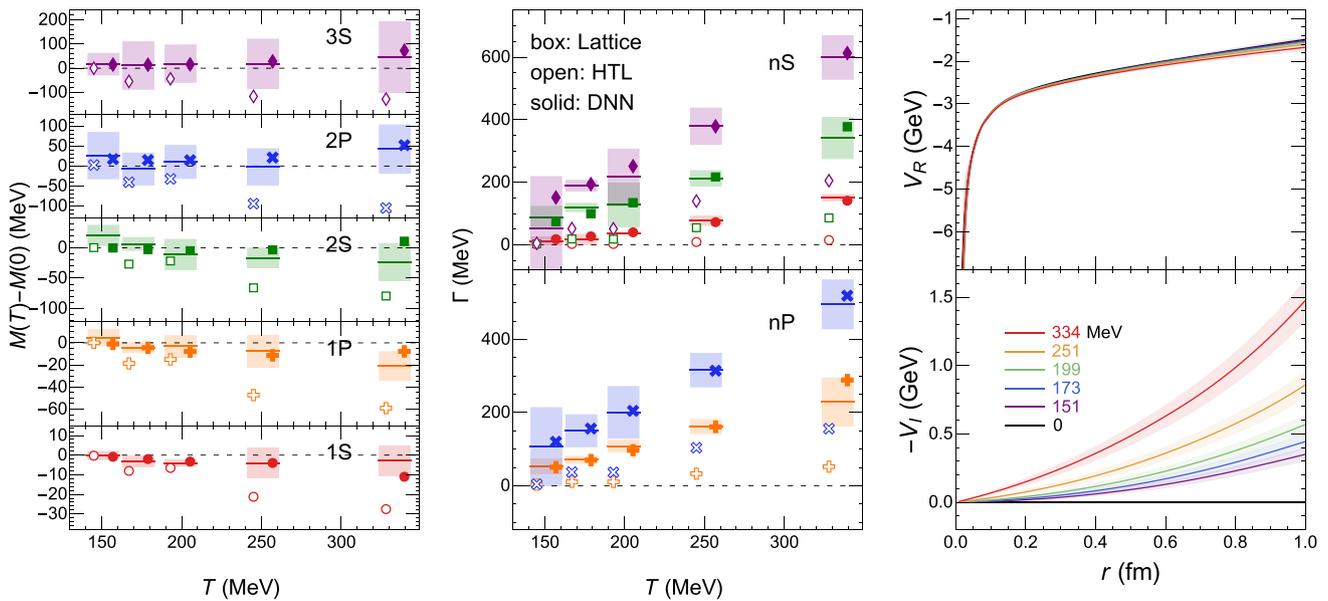


form of the heavy quark in-medium potential, i.e.,  $V_R(T, r)$  and  $V_I(T, r)$ . Note that the mass shift may affect the quarkonium production in HICs [141].

In Ref. [124], the authors developed a model-independent DNN-based method for reconstructing the temperature and inter-quark distance-dependent in-medium heavy quark potential according to the aforementioned lattice QCD results for bottomonium. Inspired by the universal approximation theorem, the authors introduced the DNN to parameterize the potential in an unbiased yet flexible manner (can be named as potential-DNN). The DNN-represented heavy quark potential is coupled to the Schrödinger equation solving process to be converted into complex valued energy eigenvalues  $E_n$ , which are related to the bound state in-medium mass and thermal width through  $\text{Re}[E_n] = m_n - 2m_b$  and  $\text{Im}[E_n] = -\Gamma_n$ . Through comparison with the lattice QCD “measurements”, the corresponding  $\chi_2$  provide the loss function for optimizing the parameters of the potential-DNN:

$$\mathcal{L} = \frac{1}{2} \sum_{T,n} \left( \frac{m_{T,n} - m_{T,n}^{\text{LQCD}}}{\delta m_{T,n}^{\text{LQCD}}} \right)^2 + \left( \frac{\Gamma_{T,n} - \Gamma_{T,n}^{\text{LQCD}}}{\delta \Gamma_{T,n}^{\text{LQCD}}} \right)^2, \quad (47)$$

with  $T \in \{0, 151, 173, 199, 251, 334\}$  MeV and  $n \in \{1S, 2S, 3S, 1P, 2P\}$  according to the lattice QCD evaluation conditions. Gradient descent with backpropagation can be applied for the DNN optimization here, where the gradient is estimated efficiently from perturbative analysis based on the Schrödinger equation with respect to the perturbative change of the potential and just arrived at the Hellman–Feynman theorem. Furthermore, the uncertainty of the reconstructed potential is quantified via Bayesian inference; thus, the posterior distribution of the DNN parameters is evaluated. With the outlined approach, in Ref. [124], good agreement with the lattice QCD results for the masses and thermal widths of bottomonium was achieved simultaneously; see the left and middle panels of Fig. 13. Additionally, the temperature- and distance-dependent heavy quark potential was obtained, as shown in the right panel of Fig. 13. Clearly, the color screening effect emerged for the reconstruction with a flatter structure appearing in  $V_R(T, r)$  with



**Fig. 13** (Color online) Picture taken from Ref. [124]. Left and middle: In-medium mass shifts with respect to the vacuum mass (left) and the thermal widths (right) of different bottomonium states obtained from fits to LQCD results of Ref. [125] (lines and shaded bands) using weak-coupling-motivated functional forms [126] (open symbols) and DNN-based optimization (solid symbols). The points are shifted horizontally for better visualization.  $Y(1S)$ ,  $\chi_{b_0}(1P)$ ,

$Y(2S)$ ,  $\chi_{b_0}(2P)$ , and  $Y(3S)$  states are represented by red circles, orange pluses, green squares, blue crosses, and purple diamonds, respectively. Right: The DNN-reconstructed real (top) and imaginary (bottom) parts of the heavy quark potential at temperatures of  $T = 0$  (black), 151 (purple), 173 (blue), 199 (green), 251 (orange), and 334 MeV (red). The uncertainty bands represent the 68%(1 $\sigma$ ) confidence region

the increasing temperature at a long distance, but the temperature dependence was mild compared with the perturbative analysis-based results in the same temperature range. In contrast, the imaginary part, i.e.,  $V_I(T, r)$ , exhibited significant growth with respect to both temperature and distance and also exhibited a larger magnitude than the one-loop HTL-motivated results.

### 5.3 Deep learning for quasi-particle mass

The EoS of hadron resonance gas in the QCD phase diagram can be calculated using a simple statistical formula with the following partition function:

$$\ln Z(T) = \sum_i \ln Z_{hi}(T), \tag{48}$$

where  $Z_{hi}(T)$  is the partition function for one of the several hundred hadrons in HRG, assuming that there is no interaction between different hadrons. The obtained EoS agrees with lattice QCD calculations. It is impossible to obtain the lattice QCD EoS for QGP using the same formula, as quarks and gluons interact with each other and form a many-body quantum system. However, if one assumes that the quarks and gluons are non-interacting quasi-particles whose masses

depend on the local temperature, the lattice QCD EoS can be reproduced using the following simple statistical formula:

$$\begin{aligned} \ln Z(T) &= \ln Z_g(T) + \sum_i \ln Z_{q_i}(T) \\ \ln Z_g(T) &= -\frac{d_g V}{2\pi^2} \int_0^\infty p^2 dp \\ &\quad \ln \left[ 1 - \exp \left( -\frac{1}{T} \sqrt{p^2 + m_g^2(T)} \right) \right] \\ \ln Z_{q_i}(T) &= +\frac{d_{q_i} V}{2\pi^2} \int_0^\infty p^2 dp \\ &\quad \ln \left[ 1 + \exp \left( -\frac{1}{T} \sqrt{p^2 + m_{q_i}^2(T)} \right) \right], \end{aligned} \tag{49}$$

where  $Z_g$  represents the partition function of quasi-gluons;  $Z_{q_i}$  represents the partition function of quasi-quarks;  $d_g$  and  $d_{q_i}$  represent the spin and color degeneracy for gluons and quarks, respectively;  $p$  represents the magnitude of momentum; and  $T$  represents the local temperature. Gluons, along with up, down, and strange quarks, are considered in this calculation. It is assumed that the temperature quasi-particle masses  $m_{u/d}(T)$  are the same for up and down quarks but different for gluons  $m_g(T)$  and strange quarks  $m_s(T)$ . Thus, there are three variational functions whose forms are unknown and must be determined by mapping the following EoS to the lattice QCD EoS:

$$\begin{aligned}
 P(T) &= T \left( \frac{\partial \ln Z(T)}{\partial V} \right)_T \\
 \epsilon(T) &= \frac{T^2}{V} \left( \frac{\partial \ln Z(T)}{\partial T} \right)_V
 \end{aligned}
 \tag{50}$$

Several deep residual neural networks were constructed to represent the variational functions  $m_{u/d}(T)$ ,  $m_s(T)$ , and  $m_g(T)$ . The mass functions of these quasi-partons are used in Eq. 49 to compute the partition function. The resulting partition function is used in Eq. 50 to compute the pressure and energy density as a function of the temperature. This procedure involves both numerical integration and differentiation. The integration is implemented using Gaussian quadrature with the TensorFlow library, while the differentiation is given by auto-differentiation. The loss function is designed as

$$\text{loss} = |s_{\text{dnn}} - s_{\text{lattice}}|^2 + |\Delta_{\text{dnn}} - \Delta_{\text{lattice}}|^2 + L_{\text{constrain}}, \tag{51}$$

where  $s = (\epsilon + P)/T$  represents the entropy density, and  $\Delta = (\epsilon - 3P)/T^4$  represents the trace anomaly. The  $L_{\text{constrain}}$  contains physical constraints in the high-temperature region whose theoretical function form is given by HTL calculations. The learned quasi-partons reproduce the lattice QCD EoS. Using these mass functions, the authors calculated  $\eta/s(T)$  and found that its minimum was located at approximately  $1.25T_c$  [142].

## 6 Hard probe

Energetic partons lose energy as they pass through the hot QGP. This process is quantified by the jet transport coefficient  $\hat{q}$ , which is defined as the transverse momentum broadening squared per unit length [143–147]. The temperature-dependent jet transport coefficient for heavy quarks was extracted using Bayesian analysis with the D-meson  $v_2$  and  $R_{AA}$  data from different experiments [148]. Bayesian inference was used to extract the jet energy loss distributions, and the observed jet quenching was dominated by a few out-of-cone scatterings [149]. The JETSCAPE collaboration extracted  $\hat{q}$  with a multi-stage jet evolution model [150]. In these studies, parametrized forms were typically used for the unknown  $\hat{q}(T)$  function. An information field is proposed to provide nonparametric functions for global Bayesian inference to avoid long-range correlations and human biases [151, 152].

Deep learning has been widely used in high-energy particle physics to analyze the substructures of jets and to classify jets using the momentum of final-state hadrons in jets [153, 154]. In HICs, deep learning is used not only to classify quark and gluon jets but also to study the jet energy loss,

the medium response, and the initial jet production positions [155–157].

Constraining the initial jet production positions will allow more detailed and differential studies of jet quenching. For example, one task in the field of HICs is to search for Mach cones in QGP produced by the supersonic parton jets. The difficulty is that the jets are produced at different locations in the initial state and travel in different directions in the QGP. Consequently, the shape of the Mach cone depends on the path length and is distorted by the local radial flow and temperature gradient. Predicting jet production positions using deep learning will help to select jet events whose Mach cones have similar shapes, enhancing the signal of the Mach cones in the final-state hadron distribution.

In these studies, the training data are usually generated by jet transport models [158, 159]; e.g., in the linear Boltzmann transport model (LBT), the jet parton loses energy through elastic scattering with thermal partons in QGP and inelastic gluon radiation. This process is described by a linearized Boltzmann equation:

$$\begin{aligned}
 p_a \cdot \partial f_a &= \int \prod_{i=b,c,d} \frac{d^3 p_i}{2E_i(2\pi)^3} \frac{\gamma_b}{2} (f_{cd} - f_{ab}) |\mathcal{M}_{ab \rightarrow cd}|^2 \\
 &\times S_2(\hat{s}, \hat{t}, \hat{u})(2\pi)^4 \delta^4(p_a + p_b - p_c - p_d) + \text{inelastic}.
 \end{aligned}
 \tag{52}$$

where  $f_{a/c}$  are the distribution functions of the jet partons before and after scattering in the forward process, and  $f_{b/d} = 1/[e^{\frac{p \cdot u}{T}} \pm 1]$  are the Fermi–Dirac and Bose–Einstein distributions for thermal quarks and gluons, respectively, in QGP. On the right-hand side,  $f_c f_d$  corresponds to the gain term and  $-f_a f_b$  corresponds to the loss term of elastic scattering, whose amplitude is squared as  $|\mathcal{M}_{ab \rightarrow cd}|^2$  from leading-order perturbative QCD calculations.  $\gamma_b$  represents the color and spin degeneracy of the thermal parton  $b$ , and the term  $\hat{S}_2 = \theta(\hat{s} > 2\mu_D^2)\theta(-\hat{s} + \mu_D^2 \leq \hat{t} \leq -\mu_D^2)$  is used to regularize the collinear divergence. The inelastic part comes from the gluon radiation described by higher-twist calculations.

The lost energy is deposited in QGP as represented by source terms of the relativistic hydrodynamic equations:

$$\nabla_\mu T^{\mu\nu} = J^\nu, \tag{53}$$

where  $T^{\mu\nu}$  represents the local energy momentum tensor of the QGP and  $J^\nu$  is the source term. In practice, if the energy deposited on the recoiled thermal parton exceeds 2 GeV, it is removed and placed into the LBT. This leaves a negative jet source in the QGP. If the deposited energy is less than 2 GeV, this corresponds to a positive jet source. Recoiled partons in the LBT do not interact with each other, which explains why the LBT solves a linearized Boltzmann equation. Recently, the LBT was extended to the QLBT, which

treats quarks and gluons as quasi-partons to constrain various transport parameters [160].

The initial jet production positions are sampled from the distribution of hard scattering, which is proportional to the distribution of binary collisions. The initial entropy density distribution is provided by the TRENTo Monte Carlo model, from which the initial  $T^{\mu\nu}$  can be calculated. Simultaneously solving Eqs. 52 and 53 provides both the jet energy loss and the medium response in each simulation. Typically, 10 ~ 100 thousands jet events are needed to predict the initial jet production positions. Of course, a larger amount of training data is better, provided that there are sufficient computational resources.

One may ask whether there is a type of DNN that is best suited to studying jet energy loss and predicting jet production positions. In practice, CNNs, point cloud neural networks, and graph neural networks have been used in different projects. Typically, the performance of different neural-network architectures is tested, and the one that works best for the specific task is selected. The simplest yet most powerful CNN should be the first to be tested in jet shape and jet energy loss studies. To capture the full information in jets, a point cloud network and a message-passing neural network can be used.

## 7 Observables in HICs

### 7.1 PCA for flow analysis

In relativistic HICs, the collective flow provides important information about the properties of the QGP and its initial state fluctuations [95–100]. The flow observables are generally defined by a Fourier decomposition of the produced particle distribution in the momentum space, such as

$$\frac{dN}{d\phi} = \frac{1}{2\pi} \sum_{-\infty}^{\infty} \vec{V}_n e^{-in\phi} = \frac{1}{2\pi} (1 + 2 \sum_{n=1}^{\infty} v_n e^{-in(\phi - \Psi_n)}), \quad (54)$$

where  $\vec{V}_n = v_n e^{in\Psi_n}$  is the flow vector of order  $n$ ,  $v_n$  represents flow harmonics of order  $n$ , and  $\Psi_n$  represents the corresponding event plane angle. Additionally, the flow coefficients can be obtained from the two-particle correlations associated with a Fourier decomposition:

$$\left\langle \frac{dN_{\text{pairs}}}{dp_1 dp_2} \right\rangle \propto 1 + 2 \sum_{n=1}^{\infty} V_{n\Delta}(p_{T1}, p_{T2}) \cos(n\Delta\phi), \quad (55)$$

where  $V_{n\Delta}(p_{T1}, p_{T2})$  is a symmetric covariance matrix and  $\Delta\phi = \phi^a - \phi^b$  represents the relative azimuthal angle between two emitted particles. Under the assumption of flow factorization,  $V_{n\Delta}(p_{T1}, p_{T2})$  is related to the flow harmonics  $v_n(p_T)$  as follows:  $V_{n\Delta}(p_{T1}, p_{T2}) \approx v_n(p_{T1})v_n(p_{T2})$  [161] (For

other flow methods and flow measurements, see [97, 100, 162–164]).

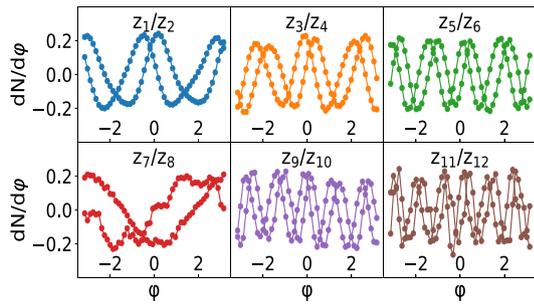
Recently, an ML technique called PCA based on SVD has been used to study the collective flow in relativistic HICs. For the two-particle correlations with the Fourier expansion [166–169], the event-by-event flow fluctuations have been investigated via PCA, revealing the substructures of the flow fluctuations [166–168]. Using PCA,  $V_{n\Delta}(p_{T1}, p_{T2})$  can be expressed as [167]

$$V_{n\Delta}(p_{T1}, p_{T2}) = \sum_{\alpha} v_n^{(\alpha)}(p_{T1})v_n^{(\alpha)}(p_{T2}), \quad (56)$$

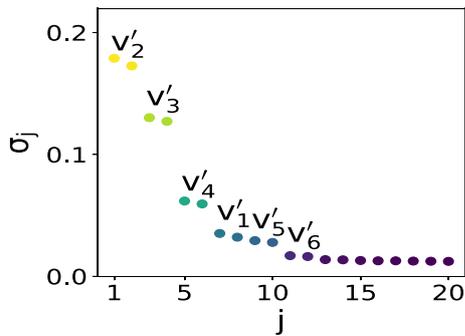
$$\text{with } \int dp_T w^2(p_T) v_n^{(\alpha)}(p_T) v_n^{(\beta)}(p_T) = \lambda_{\alpha} \delta_{\alpha\beta} \quad (57)$$

where  $v_n^{(\alpha)}(p_T)$  are the eigenvectors of the two-particle covariance matrix, and  $w(p_T)$  is the weight for the particle.  $\alpha = 1$  denotes the leading modes,  $\alpha = 2$  denotes the subleading modes,  $\alpha = 3$  denotes the subsubleading modes, and so on. It was found that the leading modes correspond to the traditional flow harmonics and that the subleading modes lead to the breakdown of the flow factorization. In Ref. [167, 168], a linear relationship  $V_n^{(\alpha)} \propto \xi_n^{(\alpha)}$  was demonstrated for the leading, subleading, and subsubleading modes via hydrodynamic simulations. In Ref. [169], PCA was used to study the mode coupling between flow harmonics, which revealed hidden mode-mixing patterns that had not been previously discovered. Recently, the CMS collaboration extracted the subleading flow modes for Pb+Pb and p+Pb collisions at the LHC, reporting qualitative agreement between experimental measurements and theoretical calculations [170]. Using AMPT and HIJING simulations, Ref. [171] showed that the PCA modes depend on the choice of the  $p_T$  range and the particle weight  $w$ . In addition, the leading modes are influenced by non-flow effects, and the mixing between the non-flow and leading flow modes leads to fake subleading modes. Therefore, it is important to carefully handle the non-flow effects and the choice of the weight and phase space when implementing PCA to extract the subleading flow modes in both experimental and theoretical studies.

The aforementioned PCA studies on collective flow [166–171] were all based on the correlation data obtained with a Fourier expansion. Recently, PCA has been applied directly to single particle distributions  $dN/d\phi$  without prior treatment with a Fourier transform, for exploring whether it can be used to directly discover flow without the guidance from humans [165]. Specifically, with PCA matrix multiplication, the  $i_{\text{th}}$  row of a particle distribution matrix with  $N$  events generated from VISH2+1 hydrodynamics can be expressed as



**Fig. 14** (Color online) PCA eigenvectors  $z_j$  for the final-state matrix of particle distributions, generated from VISH2+1 hydrodynamics in 2.76 A TeV Pb+Pb collisions at 10–20% centrality [165]



**Fig. 15** (Color online) Singular values of PCA for the final-state matrix of particle distributions in Pb+Pb collisions at 10–20% centrality [165]

$$\begin{aligned}
 dN/d\phi^{(i)} &= \sum_{j=1}^m x_j^{(i)} \sigma_j z_j = \sum_{j=1}^m \tilde{v}_j^{(i)} z_j \\
 &\approx \sum_{j=1}^k \tilde{v}_j^{(i)} z_j \quad (i) = 1, \dots, N
 \end{aligned}
 \tag{58}$$

Here,  $(i) = 1, 2, \dots, N$  is the index of the event.  $j$  is the index for the azimuthal angle, where the total azimuthal angle  $[-\pi, \pi]$  is divided into  $m$  bins to count the particles in each bin. After SVD,  $dN/d\phi^{(i)}$  can be expressed by a linear combination of the eigenvectors  $z_j$  with the corresponding coefficient  $\tilde{v}_j^{(i)}$  (where  $j = 1, 2, \dots, m$ ), and  $\sigma_j$  represents the diagonal

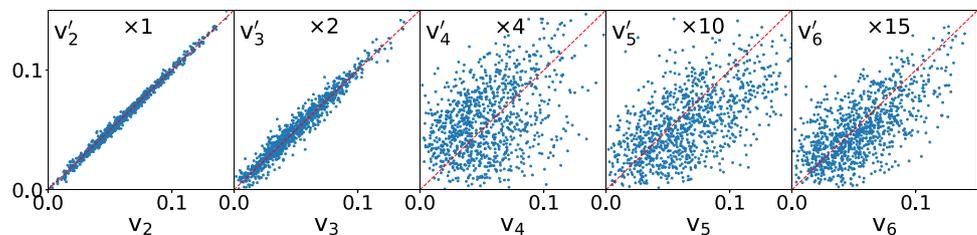
elements (singular values) of the particle distribution matrix, which are arranged in descending order. In the spirit of PCA, in the last step, a cut is made at the indices  $k$  to focus only on the most important components.

Figures 14 and 15 show the first 12 eigenvectors  $z_j$  and the first 20 singular values  $\sigma_j$  of the PCA in descending order for the final-state matrix constructed from 2000  $dN/d\phi$  distributions with the azimuthal angle  $[-\pi, \pi]$  equally divided into 50 bins. Such  $dN/d\phi$  distributions are generated from the VISH2+1 hydrodynamics with event-by-event fluctuating TRENTo initial conditions for 2.76 A TeV Pb+Pb collisions at 10–20% centrality. Figure 14 shows that the PCA eigenvectors are similar to the traditional Fourier bases. For example, the 1st and 2nd eigenvectors are close to  $\sin(2\phi)$  and  $\cos(2\phi)$ , and the 3rd and 4th eigenvectors are close to  $\sin(3\phi)$  and  $\cos(3\phi)$ . The corresponding singular values in Fig. 15 are arranged in pairs, which correspond to the real and imaginary parts of the anisotropic flow. It was found that for  $n \leq 6$ , the values of these PCA flow harmonics were very close to those of the traditional event-averaged flow harmonics obtained from the Fourier expansion but not exactly the same. Figure 16 presents a comparison of the event-by-event flow harmonics obtained from PCA and from the traditional Fourier expansion. As shown, the elliptic flow with  $n = 2$  and the triangular flow with  $n = 3$  from the two methods agreed well. However, for higher flow harmonics with  $n \geq 4$ , the PCA and Fourier expansion results differed significantly owing to the mode-mixing effects. In Ref. [165], with these PCA flow harmonics  $v'_n$ , the symmetric cumulants  $SC'(m, n)$  were calculated. Except for  $SC'(2, 3)$ , these PCA symmetric cumulants were significantly reduced compared with the traditional Fourier ones, because of the significantly increased linearity between the PCA flow harmonics and the initial eccentricities. These results indicated that PCA could define the collective flow on its own basis. Compared with the traditional ones obtained from the Fourier decomposition, the PCA method reduces the mode coupling effects between different flow harmonics [165].

### 7.2 CME detection

In the presence of a magnetic field, the chiral magnetic effect (CME) can occur when the system has a chiral imbalance, i.e., the numbers of left- and right-handed particles differ.

**Fig. 16** (Color online) Comparison between the event-by-event flow harmonics  $v'_n$  from PCA and  $v_n$  from the Fourier expansion in Pb+Pb collisions at 10–20% centrality [165]



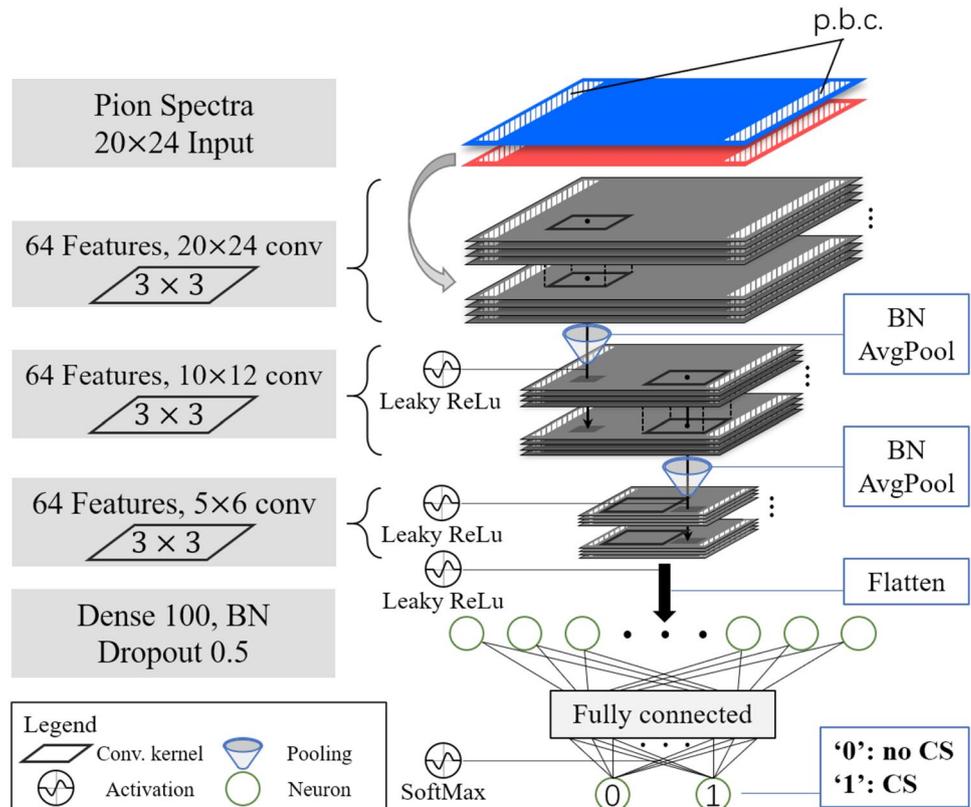
Essentially, a current of electric charge (known as chiral magnetic current) can be induced to flow along the direction of the magnetic field. The use of the CME to reveal the vacuum structure of QCD has been proposed. In HICs, a strong magnetic field can be created by the motion of the colliding ions, and it is predicted that in the formed hot and dense QGP, the topological fluctuations of gluon fields may cause chiral imbalance for quarks. Accordingly, the CME may occur, which can manifest as a separation of electric charge along the magnetic-field direction. However, several challenges hinder the detection of the CME in HICs, among which the chief difficulty is disentangling the CME signal from other possible sources of charge separation (CS), e.g., elliptic flow, the global polarization, and other background noises, although multiple observables are proposed.

Despite the challenges, there is long-term and continuing interest in the search for the CME in HICs because of its general importance to QCD. Recently, Ref. [172] proposed the use of deep learning to construct an end-to-end CME-meter that can efficiently analyze the final-state hadronic spectrum as a whole in the sense of Big Data with a deep CNN to reveal the fingerprints of the CME. For supervised learning, the training set was prepared from the string melting AMPT model with CME implemented under a global CS scheme. Essentially, the CME events are generated by switching the y-components of momenta of a fraction of a

downward moving light quark and its corresponding anti-quarks with upward moving direction. The fraction defines the CS fraction  $f$ , which separates the events into the “no CS” (label as “0”) class for those with  $f = 0\%$  and the “CS” class (label as “1”) for those with  $f > 0\%$ . Each event is represented as 2D transverse momentum and azimuthal angle spectra of charged pions in the final state, i.e.,  $\rho_\pi(p_T, \phi)$ . Then, the deep CNN is trained to perform binary classification on the labeled events with the spectra to be the input. Figure 17 shows the architecture of the developed deep CNN for CME-meter construction.

As shown in Fig. 17, the output of the network has two nodes, each of which is naturally interpreted as the probability resulting from the network decision in recognizing any given input spectrum as CME ( $P_1$ ) or non-CME ( $P_0 = 1 - P_1$ ) events. The training set contains multiple collision beam energies and centralities for diversity consideration. The pion spectrum is obtained by averaging over 100 events with the same collision condition to reduce the fluctuations, which also reduces the backgrounds and thus should be considered a prerequisite for realistic application in experiments. For the training, different levels of the CS fraction are used, and it is found that the classification validation accuracy is lower for a smaller number of CS fraction training events. This indicates that a larger CS fraction can be identified more easily, which is expected. Despite

**Fig. 17** (Color online) Taken from [172]. The CNN architecture with  $\pi^+$  and  $\pi^-$  spectra  $\rho^\pm(p_T, \phi)$  as inputs



the different induced discernibility, the trained deep CNNs all exhibited robust performance against varying collision centrality and energy. One can conclude that at least under the AMPT modeling level, the CS signals can survive into the final state of the collision dynamics at different collision conditions, which can be recognized by the deep CNN-based CME-meter.

Note that the network was trained only on Au+Au collision systems, while the extrapolation to other collision system was validated. Specifically, the obtained CME-meter was applied to isobaric collisions of  $^{96}_{40}\text{Zr}+^{96}_{40}\text{Zr}$  and  $^{96}_{44}\text{Ru}+^{96}_{44}\text{Ru}$ , which were proposed for the search of the CME. Because Ru contains more protons than Zr, which induce a stronger magnetic field, there is expected to be a stronger CS signal in Ru+Ru collisions. To reveal this difference and also the distinguishable difference of the CME-meter for the two isobaric collision systems from  $P_1^{\text{Ru}} > P_1^{\text{Zr}}$ , the  $R_{\text{iso}}$  was evaluated, and the results validated the developed CNN-based CME-meter:

$$R_{\text{iso}} = 2 \times \frac{\langle \text{logit}(P_1^{\text{Ru}}) \rangle - \langle \text{logit}(P_1^{\text{Zr}}) \rangle}{\langle \text{logit}(P_1^{\text{Ru}}) \rangle + \langle \text{logit}(P_1^{\text{Zr}}) \rangle}, \tag{59}$$

where the function  $\text{logit}(x) = \log[x/(1 - x)]$  is used to restore the derivative in the saturation region of the activation in the last layer of the NN, i.e., softmax. From Tab. 1 on  $R_{\text{iso}}$ , the trained CME-meter was well validated beyond the training collision system, indicating its robust capture of the general CME signal in the collisions.

The CME-meter was also validated through a different model simulation, i.e., anomalous-viscous fluid dynamics (AVFD).  $P_1$  exhibited a consistent positive correlation with  $N_5/S$ , which controlled the CME strength, while the contamination from local charge conservation (LCC) up to 30% did not augment the performance of the CME-meter on the testing events from AVFD. In Ref. [172], to reveal the underlying account for the trained CME-meter, the network output  $P_1$  and  $\gamma$ -correlator were compared. The  $\gamma$ -correlator—a conventional CME probe—can measure the event-by-event two-particle azimuthal correlation of charged hadrons. It was shown that for averaged events, both the CME signal and the background from  $\delta\gamma$  (difference between correlations within particles of the same charge and within particles of opposite charge) are suppressed. Being differently, the CME-meter output  $P_1$  works well in classifying CS and no-CS classes on the averaged events.

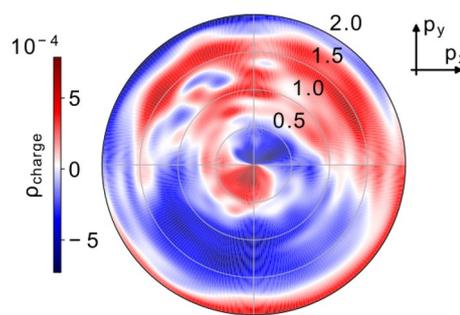
The direct implementation of this trained CME-meter in real experiments would require reconstructing the reaction plane of each collision event to form the averaged events as input for the meter. In general, the reaction plane reconstruction can be achieved by measuring correlations of final-state particles, and it inevitably contains finite resolution and background effects. It was shown that even in restricted event plane reconstruction, the trained CME-meter can recognize the CS signals. For the deployment of the trained CME-meter on single event measurements, Ref. [172] proposed a hypothesis test perspective.

Another way to interpret the trained deep-learning algorithm is the DeepDream method, which was used in Ref. [172] to reconstruct the network most responding input pion spectrum, manifesting the ‘‘CME pattern’’ that the CNN-based CME-meter essentially captured for its further CME signal recognition. The key idea is to perform variational tuning on the input pion spectrum with the trained and frozen network to maximize its output (i.e., pushing  $P_1 \rightarrow 1$ ), driven by the gradient  $\delta P_1(\rho_\pi(p_T, \phi))/\delta \rho_\pi(p_T, \phi)$ . The resultant ‘‘CME pattern’’ from the trained network is displayed in Fig. 18, where the charge conservation and a clear dipole structure appear, both being CME-related features.

## 8 Summary and outlook

### 8.1 Summary

As a modern computational paradigm, AI—particularly machine- and deep-learning techniques—has introduced a wealth of applications and new possibilities in scientific research. Owing to its special ability to recognize patterns



**Fig. 18** (Color online) *DeepDream* map for the (0% + 10%) model [172]

**Table 1** The results of the (0% + 10%) model on the isobaric collision systems (Ru+Ru and Zr+Zr at 200 GeV)

Centrality	0–10%	10–20%	20–30%	30–40%	40–50%	50–60%
$R_{\text{iso}}$	9.95%	12.99%	8.13%	13.84%	19.67%	10.47%

and structures hidden in complex data, these learning-based strategies make physics exploration with a Big Data or smart computation mindset feasible. In the context of HENP revolving around HIC programs to understand nuclear matter properties under different conditions, various research fields have benefited from the incorporation of these techniques.

In this mini-review, we presented the recent progress in the field of HICs, including initial state physics inference, QCD matter transport and bulk properties, thermal medium modifications for partons or hadrons, and recognition of physical observables in HICs.

We first reviewed different loss functions  $l$  used in supervised learning, unsupervised learning, semi-supervised learning, self-supervised learning, and active learning. During training, the negative gradients  $-\frac{\partial l}{\partial \theta}$  are used to optimize the network in SGD-like algorithms. AD is employed to efficiently compute the derivatives of the loss with regard to model parameters  $\theta$ . As auto-diff has analytical precision for the variational function represented by the neural network, it has been widely used in physics-informed neural networks to solve ODEs and PDEs. We then introduced the widely used neural-network architectures, such as the MLP, CNN, RNN, and point cloud network. Next, we explained the generative models, such as the autoencoder, GAN, flow model, and diffusion models, in detail. These models are widely used in lattice QCD to generate field configurations.

For the initial condition, ML has been widely used to determine the centrality classes and impact parameters using the final-state hadrons in the momentum space, for extracting the initial nuclear structures, such as the nuclear deformation, the  $\alpha$  clustering, and the neutron skin. In general, it is easier to extract the nuclear deformation than the  $\alpha$  clustering and neutron skin according to the current literature.

For bulk matter, Bayesian parameter estimation has been successfully used to determine the temperature-dependent shear and bulk viscosities of QGP. An unsupervised autoencoder was used to reconstruct the charged multiplicity distributions, which helps to determine the source temperature and the temperature of the nuclear liquid gas phase transition. Deep CNNs, point cloud networks, and event-averaging techniques are employed to classify the crossover and first-order phase transition regions in the QCD phase diagram, using data generated by relativistic hydrodynamic models and hadronic transport models. Active learning is used to map out thermodynamically unstable regions near the critical endpoint in the QCD phase diagram. For hydrodynamic evolution, a well-designed network called sU-net can capture the nonlinear mapping between the initial and final profiles with sufficient precision, which is also far faster than the traditional hydrodynamic simulations.

For QGP in-medium effects, we first reported some of the recently proposed ML-based methods for spectral function reconstruction, which is a notorious ill-posed inverse

problem. Both supervised and unsupervised methods have been discussed for the inference of spectral out-of-Euclidean correlator measurements from Monte Carlo simulations (e.g., lattice study). Then, in-medium heavy quark interaction inference based on in-medium heavy quarkonium spectroscopy was introduced. A novel DNN representation integrated inside the forward problem-solving pipeline with AD was proposed. This strategy is also used for in-medium quasi-particle effective model construction from the lattice QCD EoS.

For hard probes, Bayesian analysis is widely used to extract the temperature-dependent jet (or heavy quark) transport coefficient  $\hat{q}(T)$  and the jet energy loss distributions. Recently, deep learning-assisted jet tomography was developed to locate the initial jet production positions. This is important for the study of jet substructures and the medium response. Using this technique, it was observed that the signal of jet-induced Mach cones is amplified by selecting jet events.

For the observables, PCA has been implemented to study the collective flow in relativistic HICs. This revealed the substructures of the flow fluctuations, which can potentially be used to extract the subleading modes of flow with efforts from both the experimental and theoretical sides. When applied directly to the single particle distributions, PCA can directly discover flow with a basis similar to the Fourier expansion ones, which significantly reduces the mode coupling between different flow harmonics.

## 8.2 Outlook

Despite the impressive progress, the interplay between HENP and ML is still inducing hectic evolution. Many questions and challenges remain and deserve further exploration. In addition to the aforementioned applications of ML in the field of HICs, several other topics can be explored with ML, e.g., critical endpoint searching for eRHIC and the Electron Ion Collider (EIC) regime [173], spin polarization study, the upcoming FAIR program, Nuclotron-based Ion Collider facility (NICA) experiments, nuclear structure inference, and High Intensity Heavy-ion Accelerator Facility (HIAF) experiments in China. Regarding the future prospects of applying ML techniques for HIC physics research, because this field is rapidly evolving, we present questions that we consider worthy of future investigation:

- Can ML provide more efficient "observables" to pin down the desired physics?.
- Can the algorithms provide new physical knowledge to advance our understanding of nuclear matter?.
- How can we make the ML algorithms to be confronted with realistic experiments? Is on-line analysis possible?

How can experimental raw data be accessed to test neural networks pretrained with model simulations?.

- Is it possible to accelerate HIC dynamical simulations for performing high statistics measurements or Bayesian inference?.
- How can Bayesian inference be combined with ML to advance our field and better connect experiment to theory?.
- How can symmetries be fully incorporated into the analysis using ML, e.g., Lorentz Group Equivariant Autoencoders [174]? How can dimensionality analysis (constraints) be incorporated into the ML methods properly and consistently?.

It is also important to consider how we can adopt potentially useful approaches from other fields, e.g., particle physics, condensed-matter physics, and astrophysics, and how the community can better organize with joint efforts, e.g., for maximizing the potential of these novel computational techniques to advance the field of HENP.

## Declarations

**Conflict of interest** All authors declare that there are no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. L. Benato et al., Shared data and algorithms for deep learning in fundamental physics. *Comput. Softw. Big Sci.* **6**, 9 (2022). <https://doi.org/10.1007/s41781-022-00082-6>
2. M. Favoni, A. Ipp, D.I. Müller et al., Lattice gauge equivariant convolutional neural networks. *Phys. Rev. Lett.* **128**, 032003 (2022). <https://doi.org/10.1103/PhysRevLett.128.032003>
3. Z.P. Gao, Y.J. Wang, H.L. Lü et al., Machine learning the nuclear mass. *Nucl. Sci. Tech.* **32**, 109 (2021). <https://doi.org/10.1007/s41365-021-00956-1>
4. C. Xie, K. Ni, K. Han et al., Enhanced search sensitivity to the double beta decay of  $^{136}\text{Xe}$  to excited states with topological signatures. *Sci. China Phys. Mech. Astron.* **64**, 261011 (2021). <https://doi.org/10.1007/s11433-020-1693-6>
5. X.C. Ming, H.F. Zhang, R.R. Xu et al., Nuclear mass based on the multi-task learning neural network method. *Nucl. Sci. Tech.* **33**, 48 (2022). <https://doi.org/10.1007/s41365-022-01031-z>
6. X. Wu, Y. Lu, P. Zhao, Multi-task learning on nuclear masses and separation energies with the kernel ridge regression. *Phys.*

7. X.Z. Li, Q.X. Zhang, H.Y. Tan et al., Fast nuclide identification based on a sequential bayesian method. *Nucl. Sci. Tech.* **32**, 143 (2021)
8. Z. Gao, Y. Wang, Q. Li et al., Application of machine learning to study the effects of quadrupole deformation on the nucleus in heavy-ion collisions at intermediate energies. *Sci. China Phys. Mech. Astron.* **52**, 252010 (2022). <https://doi.org/10.1360/SSPMA-2021-0308>
9. P. Li, J. Bai, Z. Niu et al.,  $\beta$ -decay half-lives studied using neural network method. *Sci. China Phys. Mech. Astron.* **52**, 252006 (2022). <https://doi.org/10.1360/SSPMA-2021-0299>
10. R. Wang, Y.G. Ma, R. Wada et al., Nuclear liquid-gas phase transition with machine learning. *Phys. Rev. Res.* **2**, 043202 (2020). <https://doi.org/10.1103/PhysRevResearch.2.043202>
11. Y.D. Song, R. Wang, Y.G.M. Ma et al., Determining temperature in heavy ion collisions with multiplicity distribution. *Phys. Lett. B* **814**, 136084 (2021). <https://doi.org/10.1016/j.physletb.2021.136084>
12. W.B. He, Q.F. Li, Y.G. Ma, et al., in *Machine Learning in Nuclear Physics at Low and Intermediate Energies* (2023) [arXiv:2301.06396](https://arxiv.org/abs/2301.06396)
13. L. Ng, Å. Bibrzycki, J. Nys et al., Deep learning exotic hadrons. *Phys. Rev. D* **105**, L091501 (2022). <https://doi.org/10.1103/PhysRevD.105.L091501>
14. Z. Zhang, R. Ma, J. Hu et al., Approach the Gell–Mann–Okubo formula with machine learning. *Chin. Phys. Lett.* **39**, 111201 (2022). <https://doi.org/10.1088/0256-307X/39/11/11201>
15. K. Desai, B. Nachman, J. Thaler, Symmetry discovery with deep learning. *Phys. Rev. D* **105**, 096031 (2022). <https://doi.org/10.1103/PhysRevD.105.096031>
16. G. Goh, Why momentum really works. *Distill* (2017). <https://doi.org/10.23915/distill.00006>
17. D.P. Kingma, J. Ba, in *Adam: A Method for Stochastic Optimization* (2014) CoRR [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
18. J. Steinheimer, L. Pang, K. Zhou et al., A machine learning study to identify spinodal clumping in high energy nuclear collisions. *JHEP* **12**, 122 (2019). [https://doi.org/10.1007/JHEP12\(2019\)122](https://doi.org/10.1007/JHEP12(2019)122)
19. M. Omana Kuttan, J. Steinheimer, K. Zhou et al., Deep learning based impact parameter determination for the CBM experiment. *Particles* **4**, 47–52 (2021). <https://doi.org/10.3390/particles4010006>
20. Y.G. Huang, L.G. Pang, X. Luo, et al., in *Probing Criticality with Deep Learning in Relativistic Heavy-Ion Collisions* (2021) [arXiv:2107.11828](https://arxiv.org/abs/2107.11828)
21. D.P. Kingma, M. Welling, in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Auto-Encoding Variational Bayes. (2014) [arXiv:1312.6114v10](https://arxiv.org/abs/1312.6114v10)
22. L. Mosser, O. Dubrulle, M.J. Blunt, Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Phys. Rev. E* **96**, 043309 (2017). <https://doi.org/10.1103/PhysRevE.96.043309>
23. K. Mills, I. Tamblyn, Deep neural networks for direct, featureless learning through observation: The case of two-dimensional spin models. *Phys. Rev. E* **97**, 032119 (2018). <https://doi.org/10.1103/PhysRevE.97.032119>
24. L. de Oliveira, M. Paganini, B. Nachman, Learning Particle physics by example: location-aware generative adversarial networks for physics synthesis. *Comput. Softw. Big Sci.* **1**, 4 (2017). <https://doi.org/10.1007/s41781-017-0004-6>
25. M. Paganini, L. de Oliveira, B. Nachman, Accelerating science with generative adversarial networks: an application to 3D

- particle showers in multilayer calorimeters. *Phys. Rev. Lett.* **120**, 042003 (2018). <https://doi.org/10.1103/PhysRevLett.120.042003>
26. S. Ravanbakhsh, F. Lanusse, R. Mandelbaum, et al., in *Enabling Dark Energy Science with Deep Generative Models of Galaxy Images* (2017) [arXiv:1609.05796](https://arxiv.org/abs/1609.05796)
  27. M. Mustafa, D. Bard, W. Bhimji et al., CosmoGAN: creating high-fidelity weak lensing convergence maps using generative adversarial networks. *Comput. Astrophys. Cosmol.* **6**, 1 (2019). <https://doi.org/10.1186/s40668-019-0029-9>
  28. K. Zhou, G. Endrődi, L.G. Pang et al., Regressive and generative neural networks for scalar field theory. *Phys. Rev. D* **100**, 011501 (2019). <https://doi.org/10.1103/PhysRevD.100.011501>
  29. J.M. Pawłowski, J.M. Urban, Reducing autocorrelation times in lattice simulations with generative adversarial networks. *Mach. Learn. Sci. Tech.* **1**, 045011 (2020). <https://doi.org/10.1088/2632-2153/abae73>
  30. M. Germain, K. Gregor, I. Murray, et al., in *MADE: Masked Autoencoder for Distribution Estimation* (2015) [arXiv e-prints arXiv:1502.03509](https://arxiv.org/abs/1502.03509) (2015)
  31. A.v.d. Oord, N. Kalchbrenner, O. Vinyals, et al., in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Conditional image generation with pixelcnn decoders. NIPS'16, (Curran Associates Inc., Red Hook, NY, USA, 2016), p. 4797–4805
  32. A. Van Den Oord, N. Kalchbrenner, K. Kavukcuoglu, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, Pixel recurrent neural networks. ICML'16, (JMLR.org, 2016), p. 1747–1756
  33. A. van den Oord, S. Dieleman, H. Zen, et al., in *Wavenet: A Generative Model for Raw Audio* (2016) [cite arxiv:1609.03499](https://arxiv.org/abs/1609.03499)
  34. L. Wang, Y. Jiang, L. He et al., Continuous-mixture autoregressive networks learning the Kosterlitz–Thouless transition. *Chin. Phys. Lett.* **39**, 120502 (2022). <https://doi.org/10.1088/0256-307X/39/12/120502>
  35. L. Dinh, D. Krueger, Y. Bengio, in *NICE: Non-linear Independent Components Estimation* (2014) [arXiv e-prints arXiv:1410.8516](https://arxiv.org/abs/1410.8516)
  36. D. Jimenez Rezende, S. Mohamed, in *Variational Inference with Normalizing Flows* (2015) [arXiv e-prints arXiv:1505.05770](https://arxiv.org/abs/1505.05770)
  37. L. Dinh, J. Sohl-Dickstein, S. Bengio, in *Density Estimation using Real NVP* (2016) [arXiv e-prints arXiv:1605.08803](https://arxiv.org/abs/1605.08803)
  38. M.S. Albergo, G. Kanwar, P.E. Shanahan, Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Phys. Rev. D* **100**, 034515 (2019). <https://doi.org/10.1103/PhysRevD.100.034515>
  39. G. Kanwar, M.S. Albergo, D. Boyda et al., Equivariant flow-based sampling for lattice gauge theory. *Phys. Rev. Lett.* **125**, 121601 (2020). <https://doi.org/10.1103/PhysRevLett.125.121601>
  40. D. Boyda, G. Kanwar, S. Racanière et al., Sampling using  $SU(N)$  gauge equivariant flows. *Phys. Rev. D* **103**, 074504 (2021). <https://doi.org/10.1103/PhysRevD.103.074504>
  41. S. Chen, O. Savchuk, S. Zheng, et al., in *Fourier-Flow Model Generating Feynman Paths* (2022) [arXiv:2211.03470](https://arxiv.org/abs/2211.03470)
  42. J. Shlens, in *A Tutorial on Principal Component Analysis* (2014) [CoRR arXiv:1404.1100](https://arxiv.org/abs/1404.1100)
  43. Y.G. Ma, S. Zhang, Influence of nuclear structure in relativistic heavy-ion collisions. In *Handbook of Nuclear Physics* ed. by I. Tanihata, H. Toki, T. Kajino, (Springer, Singapore, 2022) [https://doi.org/10.1007/978-981-15-8818-1\\_5-1](https://doi.org/10.1007/978-981-15-8818-1_5-1)
  44. P. Xiang, Y.S. Zhao, X.G. Huang, Determination of the impact parameter in high-energy heavy-ion collisions via deep learning \*. *Chin. Phys. C* **46**, 074110 (2022). <https://doi.org/10.1088/1674-1137/ac6490>
  45. F. Li, Y. Wang, H. Lü et al., Application of artificial intelligence in the determination of impact parameter in heavy-ion collisions at intermediate energies. *J. Phys. G* **47**, 115104 (2020). <https://doi.org/10.1088/1361-6471/abb1f9>
  46. L. Li, X. Chen, Y. Cui, et al., in *Fluctuation Mechanism and Reconstruction of Impact Parameter Distributions with Two-Observables for Intermediate Energy Heavy Ion Collisions* (2022) [arXiv:2201.12586](https://arxiv.org/abs/2201.12586)
  47. R.Q. Charles, H. Su, M. Kaichun, et al., in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Pointnet: Deep learning on point sets for 3d classification and segmentation (2017), p. 77–85. <https://doi.org/10.1109/CVPR.2017.16>
  48. M. Omana Kuttan, J. Steinheimer, K. Zhou et al., A fast centrality-meter for heavy-ion collisions at the CBM experiment. *Phys. Lett. B* **811**, 135872 (2020). <https://doi.org/10.1016/j.physletb.2020.135872>
  49. L.G. Pang, K. Zhou, X.N. Wang, in *Interpretable Deep Learning for Nuclear Deformation in Heavy Ion Collisions* (2019) [arXiv:1906.06429](https://arxiv.org/abs/1906.06429)
  50. Y.L. Cheng, S. Shi, Y.G. Ma, et al., in *How does Bayesian Analysis Infer the Nucleon Distributions in Isobar Collisions?* (2023) [arXiv:2301.03910](https://arxiv.org/abs/2301.03910)
  51. C. Zhang, J. Jia, Evidence of quadrupole and octupole deformations in Zr96+Zr96 and Ru96+Ru96 collisions at ultrarelativistic energies. *Phys. Rev. Lett.* **128**, 022301 (2022). <https://doi.org/10.1103/PhysRevLett.128.022301>
  52. W.B. He, Y.G. Ma, X.G. Cao et al., Giant dipole resonance as a fingerprint of clustering configurations in 12c and 16o. *Phys. Rev. Lett.* **113**, 032506 (2014). <https://doi.org/10.1103/PhysRevLett.113.032506>
  53. C.Z. Shi, Y.G. Ma,  $\alpha$ -clustering effect on flows of direct photons in heavy-ion collisions. *Nucl. Sci. Tech.* **32**, 66 (2021). <https://doi.org/10.1007/s41365-021-00897-9>
  54. S. Zhang, Y.G. Ma, J.H. Chen et al., Nuclear cluster structure effect on elliptic and triangular flows in heavy-ion collisions. *Phys. Rev. C* **95**, 064904 (2017). <https://doi.org/10.1103/PhysRevC.95.064904>
  55. J. He, W.B. He, Y.G. Ma et al., Machine-learning-based identification for initial clustering structure in relativistic heavy-ion collisions. *Phys. Rev. C* **104**, 044902 (2021). <https://doi.org/10.1103/PhysRevC.104.044902>
  56. D. Adhikari et al., Accurate determination of the neutron skin thickness of  $^{208}\text{Pb}$  through parity-violation in electron scattering. *Phys. Rev. Lett.* **126**, 172502 (2021). <https://doi.org/10.1103/PhysRevLett.126.172502>
  57. J. Xu, in *Bayesian Inference of Nucleus Resonance and Neutron Skin* (2023) [arXiv:2301.07884](https://arxiv.org/abs/2301.07884)
  58. J. Xu, W.J. Xie, B.A. Li, Bayesian inference of nuclear symmetry energy from measured and imagined neutron skin thickness in  $^{116,118,120,122,124,130,132}\text{Sn}$ ,  $^{208}\text{Pb}$ , and  $^{48}\text{Ca}$ . *Phys. Rev. C* **102**, 044316 (2020). <https://doi.org/10.1103/PhysRevC.102.044316>
  59. M.B. Tsang et al., Constraints on the symmetry energy and neutron skins from experiments and theory. *Phys. Rev. C* **86**, 015803 (2012). <https://doi.org/10.1103/PhysRevC.86.015803>
  60. S.H. Cheng, J. Wen, L.G. Cao et al., Neutron skin thickness of  $^{90}\text{Zr}$  and symmetry energy constrained by charge exchange spin-dipole excitations. *Chin. Phys. C* **47**, 024102 (2023). <https://doi.org/10.1088/1674-1137/aca38e>
  61. X.R. Huang, L.W. Chen, Supernova neutrinos as a precise probe of nuclear neutron skin. *Phys. Rev. D* **106**, 123034 (2022). <https://doi.org/10.1103/PhysRevD.106.123034>
  62. E.A. Teixeira, T. Aumann, C.A. Bertulani et al., Nuclear fragmentation reactions as a probe of neutron skins in nuclei. *Eur. Phys. J. A* **58**, 205 (2022). <https://doi.org/10.1140/epja/s10050-022-00849-w>
  63. D. Androic et al., Determination of the  $^{27}\text{Al}$  neutron distribution radius from a parity-violating electron scattering measurement.

- Phys. Rev. Lett. **128**, 132501 (2022). <https://doi.org/10.1103/PhysRevLett.128.132501>
64. H.J. Xu, in *20th International Conference on Strangeness in Quark Matter 2022, Probing neutron skin and symmetry energy with relativistic isobar collisions* (2023) [arXiv:2301.08303](https://arxiv.org/abs/2301.08303)
  65. N. Kozyrev, A. Svetlichnyi, R. Nepeivoda et al., Peeling away neutron skin in ultracentral collisions of relativistic nuclei. *Eur. Phys. J. A* **58**, 184 (2022). <https://doi.org/10.1140/epja/s10050-022-00832-5>
  66. L.M. Liu, C.J. Zhang, J. Zhou et al., Probing neutron-skin thickness with free spectator neutrons in ultracentral high-energy isobaric collisions. *Phys. Lett. B* **834**, 137441 (2022). <https://doi.org/10.1016/j.physletb.2022.137441>
  67. Y.J. Huang, L.G. Pang, X.N. Wang, Determining the neutron skin types using deep learning and nuclear collisions: an attempt. *Sci. China Phys. Mech. Astron.* **52**, 252011 (2022). <https://doi.org/10.1360/SSPMA-2021-0318>
  68. U.W. Heinz, H. Song, A.K. Chaudhuri, Dissipative hydrodynamics for viscous relativistic fluids. *Phys. Rev. C* **73**, 034904 (2006). <https://doi.org/10.1103/PhysRevC.73.034904>
  69. P. Romatschke, U. Romatschke, Viscosity information from relativistic nuclear collisions: how perfect is the fluid observed at RHIC? *Phys. Rev. Lett.* **99**, 172301 (2007). <https://doi.org/10.1103/PhysRevLett.99.172301>
  70. D. Teaney, The effects of viscosity on spectra, elliptic flow, and HBT radii. *Phys. Rev. C* **68**, 034913 (2003). <https://doi.org/10.1103/PhysRevC.68.034913>
  71. H.X. Zhang, Y.X. Xiao, J.W. Kang et al., Phenomenological study of the anisotropic quark matter in the two-flavor Nambu–Jona–Lasinio model. *Nucl. Sci. Tech.* **33**, 150 (2022). <https://doi.org/10.1007/s41365-022-01129-4>
  72. S.X. Li, D.Q. Fang, Y.G. Ma et al., Shear viscosity to entropy density ratio in the Boltzmann–Uehling–Uhlenbeck model. *Phys. Rev. C* (2011). <https://doi.org/10.1103/PhysRevC.84.024607>
  73. C.L. Zhou, Y.G. Ma, D.Q. Fang et al., Thermodynamic properties and shear viscosity over entropy-density ratio of the nuclear fireball in a quantum-molecular dynamics model. *Phys. Rev. C* **88**, 024604 (2013). <https://doi.org/10.1103/PhysRevC.88.024604>
  74. D.Q. Fang, Y.G. Ma, C.L. Zhou, Shear viscosity of hot nuclear matter by the mean free path method. *Phys. Rev. C* **89**, 047601 (2014). <https://doi.org/10.1103/PhysRevC.89.047601>
  75. X.G. Deng, P. Danielewicz, Y.G. Ma et al., Impact of fragment formation on shear viscosity in the nuclear liquid–gas phase transition region. *Phys. Rev. C* **105**, 064613 (2022). <https://doi.org/10.1103/PhysRevC.105.064613>
  76. J.E. Bernhard, J.S. Moreland, S.A. Bass et al., Applying Bayesian parameter estimation to relativistic heavy-ion collisions: simultaneous characterization of the initial state and quark-gluon plasma medium. *Phys. Rev. C* **94**, 024907 (2016). <https://doi.org/10.1103/PhysRevC.94.024907>
  77. J.E. Bernhard, J.S. Moreland, S.A. Bass, Bayesian estimation of the specific shear and bulk viscosity of quark–gluon plasma. *Nat. Phys.* **15**, 1113–1117 (2019). <https://doi.org/10.1038/s41567-019-0611-8>
  78. Z. Yang, L.W. Chen, in *Bayesian Inference of the Specific Shear and Bulk Viscosities of the Quark–Gluon Plasma at Crossover From  $\phi$  and  $\Omega$  Observables* (2022) [arXiv:2207.13534](https://arxiv.org/abs/2207.13534)
  79. M. Omana Kuttan, J. Steinheimer, K. Zhou, et al., in *The QCD EoS of Dense Nuclear Matter from Bayesian Analysis of Heavy Ion Collision Data* (2022) [arXiv:2211.11670](https://arxiv.org/abs/2211.11670)
  80. L.G. Pang, K. Zhou, N. Su et al., An equation-of-state-meter of quantum chromodynamics transition from deep learning. *Nat. Commun.* **9**, 210 (2018). <https://doi.org/10.1038/s41467-017-02726-3>
  81. L. Pang, Q. Wang, X.N. Wang, Effects of initial flow velocity fluctuation in event-by-event (3+1)D hydrodynamics. *Phys. Rev. C* **86**, 024911 (2012). <https://doi.org/10.1103/PhysRevC.86.024911>
  82. C. Shen, Z. Qiu, H. Song et al., The iEBE-VISHNU code package for relativistic heavy-ion collisions. *Comput. Phys. Commun.* **199**, 61–85 (2016). <https://doi.org/10.1016/j.cpc.2015.08.039>
  83. Z. Yang, T. Luo, W. Chen et al., 3d structure of jet-induced diffusion wake in an expanding quark-gluon plasma. *Phys. Rev. Lett.* **130**, 052301 (2023). <https://doi.org/10.1103/PhysRevLett.130.052301>
  84. G. Qin, 3d wakes on the femtometer scale by supersonic jets. *Nucl. Sci. Tech.* **34**, 22 (2023). <https://doi.org/10.1007/s41365-023-01182-7>
  85. Y.L. Du, K. Zhou, J. Steinheimer et al., Identifying the nature of the QCD transition in relativistic collision of heavy nuclei with deep learning. *Eur. Phys. J. C* **80**, 516 (2020). <https://doi.org/10.1140/epjc/s10052-020-8030-7>
  86. Y.L. Du, K. Zhou, J. Steinheimer et al., Identifying the nature of the QCD transition in heavy-ion collisions with deep learning. *Nucl. Phys. A* **1005**, 121891 (2021). <https://doi.org/10.1016/j.nuclphysa.2020.121891>
  87. J. Steinheimer, L.G. Pang, K. Zhou et al., A machine learning study on spinodal clumping in heavy ion collisions. *Nucl. Phys. A* **1005**, 121867 (2021). <https://doi.org/10.1016/j.nuclphysa.2020.121867>
  88. L. Jiang, L. Wang, K. Zhou, Deep learning stochastic processes with QCD phase transition. *Phys. Rev. D* **103**, 116023 (2021). <https://doi.org/10.1103/PhysRevD.103.116023>
  89. M. Omana Kuttan, K. Zhou, J. Steinheimer et al., An equation-of-state-meter for CBM using PointNet. *JHEP* **21**, 184 (2020). [https://doi.org/10.1007/JHEP10\(2021\)184](https://doi.org/10.1007/JHEP10(2021)184)
  90. P. Thaprasop, K. Zhou, J. Steinheimer et al., Unsupervised outlier detection in heavy-ion collisions. *Phys. Scripta* **96**, 064003 (2021). <https://doi.org/10.1088/1402-4896/abf214>
  91. Y. Wang, F. Li, Q. Li et al., Finding signatures of the nuclear symmetry energy in heavy-ion collisions with deep learning. *Phys. Lett. B* **822**, 136669 (2021). <https://doi.org/10.1016/j.physletb.2021.136669>
  92. D. Mroczek, M. Hjorth-Jensen, J. Noronha-Hostler, et al., in *Mapping out the Thermodynamic Stability of a QCD Equation of State with a Critical Point Using Active Learning* (2022) [arXiv:2203.13876](https://arxiv.org/abs/2203.13876)
  93. H. Huang, B. Xiao, Z. Liu et al., Applications of deep learning to relativistic hydrodynamics. *Phys. Rev. Res.* **3**, 023256 (2021). <https://doi.org/10.1103/PhysRevResearch.3.023256>
  94. H. Huang, B. Xiao, H. Xiong et al., Applications of deep learning to relativistic hydrodynamics. *Nucl. Phys. A* **982**, 927–930 (2019). <https://doi.org/10.1016/j.nuclphysa.2018.11.004>
  95. D.A. Teaney, in *Viscous Hydrodynamics and the Quark Gluon Plasma* (2010), p. 207–266. [https://doi.org/10.1142/9789814293297\\_0004](https://doi.org/10.1142/9789814293297_0004)
  96. P. Romatschke, New developments in relativistic viscous hydrodynamics. *Int. J. Mod. Phys. E* **19**, 1–53 (2010). <https://doi.org/10.1142/S0218301310014613>
  97. U. Heinz, R. Snellings, Collective flow and viscosity in relativistic heavy-ion collisions. *Ann. Rev. Nucl. Part. Sci.* **63**, 123–151 (2013). <https://doi.org/10.1146/annurev-nucl-102212-170540>
  98. C. Gale, S. Jeon, B. Schenke, Hydrodynamic modeling of heavy-ion collisions. *Int. J. Mod. Phys. A* **28**, 1340011 (2013). <https://doi.org/10.1142/S0217751X13400113>
  99. H. Song, Hydrodynamic modelling for relativistic heavy-ion collisions at RHIC and LHC. *Pramana* **84**, 703–715 (2015). <https://doi.org/10.1007/s12043-015-0971-2>

100. H. Song, Y. Zhou, K. Gajdosova, Collective flow and hydrodynamics in large and small systems at the LHC. *Nucl. Sci. Tech.* **28**, 99 (2017). <https://doi.org/10.1007/s41365-017-0245-4>
101. P.F. Kolb, U.W. Heinz, in *Hydrodynamic Description of Ultrarelativistic Heavy Ion Collisions* (2003), p. 634–714 [arXiv:nucl-th/0305084](https://arxiv.org/abs/nucl-th/0305084)
102. H. Song, in *Causal Viscous Hydrodynamics for Relativistic Heavy Ion Collisions*. Other thesis (2009). [arXiv:0908.3656](https://arxiv.org/abs/0908.3656)
103. H. Song, U.W. Heinz, Causal viscous hydrodynamics in 2+1 dimensions for relativistic heavy-ion collisions. *Phys. Rev. C* **77**, 064901 (2008). <https://doi.org/10.1103/PhysRevC.77.064901>
104. H. Song, U.W. Heinz, Suppression of elliptic flow in a minimally viscous quark-gluon plasma. *Phys. Lett. B* **658**, 279–283 (2008). <https://doi.org/10.1016/j.physletb.2007.11.019>
105. M.L. Miller, K. Reygers, S.J. Sanders et al., Glauber modeling in high energy nuclear collisions. *Ann. Rev. Nucl. Part. Sci.* **57**, 205–243 (2007). <https://doi.org/10.1146/annurev.nucl.57.090506.123020>
106. T. Hirano, Y. Nara, Eccentricity fluctuation effects on elliptic flow in relativistic heavy ion collisions. *Phys. Rev. C* **79**, 064904 (2009). <https://doi.org/10.1103/PhysRevC.79.064904>
107. H.J. Drescher, Y. Nara, Effects of fluctuations on the initial eccentricity from the Color Glass Condensate in heavy ion collisions. *Phys. Rev. C* **75**, 034905 (2007). <https://doi.org/10.1103/PhysRevC.75.034905>
108. H.J. Xu, Z. Li, H. Song, High-order flow harmonics of identified hadrons in 2.76A TeV Pb + Pb collisions. *Phys. Rev. C* **93**, 064905 (2016). <https://doi.org/10.1103/PhysRevC.93.064905>
109. W. Zhao, H.J. Xu, H. Song, Collective flow in 2.76 A TeV and 5.02 A TeV Pb+Pb collisions. *Eur. Phys. J. C* **77**, 645 (2017). <https://doi.org/10.1140/epjc/s10052-017-5186-x>
110. J.S. Moreland, J.E. Bernhard, S.A. Bass, Alternative ansatz to wounded nucleon and binary collision scaling in high-energy nuclear collisions. *Phys. Rev. C* **92**, 011901 (2015). <https://doi.org/10.1103/PhysRevC.92.011901>
111. H. Yoon, J.H. Sim, M.J. Han, Analytic continuation via domain knowledge free machine learning. *Phys. Rev. B* **98**, 245101 (2018). <https://doi.org/10.1103/PhysRevB.98.245101>
112. R. Fournier, L. Wang, O.V. Yazyev et al., Artificial neural network approach to the analytic continuation problem. *Phys. Rev. Lett.* **124**, 056401 (2020). <https://doi.org/10.1103/PhysRevLett.124.056401>
113. L. Kades, J.M. Pawłowski, A. Rothkopf et al., Spectral reconstruction with deep neural networks. *Phys. Rev. D* **102**, 096001 (2020). <https://doi.org/10.1103/PhysRevD.102.096001>
114. L. Wang, S. Shi, K. Zhou, Reconstructing spectral functions via automatic differentiation. *Phys. Rev. D* **106**, L051502 (2022). <https://doi.org/10.1103/PhysRevD.106.L051502>
115. J. Horak, J.M. Pawłowski, J. Rodríguez-Quintero et al., Reconstructing QCD spectral functions with Gaussian processes. *Phys. Rev. D* **105**, 036014 (2022). <https://doi.org/10.1103/PhysRevD.105.036014>
116. J. Horak, J. Papavassiliou, J.M. Pawłowski et al., Ghost spectral function from the spectral Dyson-Schwinger equation. *Phys. Rev. D* (2021). <https://doi.org/10.1103/PhysRevD.104.074017>
117. A.K. Cyrol, J.M. Pawłowski, A. Rothkopf et al., Reconstructing the gluon. *SciPost Phys.* **5**, 065 (2018). <https://doi.org/10.21468/SciPostPhys.5.6.065>
118. L. Wang, S. Shi, K. Zhou, in *35th Conference on Neural Information Processing Systems*, Automatic differentiation approach for reconstructing spectral functions with neural networks (2021). [arXiv:2112.06206](https://arxiv.org/abs/2112.06206)
119. S. Shi, L. Wang, K. Zhou, Rethinking the ill-posedness of the spectral function reconstruction—why is it fundamentally hard and how Artificial Neural Networks can help. *Comput. Phys. Commun.* **282**, 108547 (2023). <https://doi.org/10.1016/j.cpc.2022.108547>
120. S. Soma, L. Wang, S. Shi et al., Neural network reconstruction of the dense matter equation of state from neutron star observables. *JCAP* **08**, 071 (2022). <https://doi.org/10.1088/1475-7516/2022/08/071>
121. S. Soma, L. Wang, S. Shi, et al., in *Reconstructing the Neutron Star Equation of State from Observational Data via Automatic Differentiation* (2022) [arXiv:2209.08883](https://arxiv.org/abs/2209.08883)
122. X. Gao, A.D. Hanlon, N. Karthik et al., Continuum-extrapolated NNLO valence PDF of the pion at the physical point. *Phys. Rev. D* **106**, 114510 (2022). <https://doi.org/10.1103/PhysRevD.106.114510>
123. M. Zhou, F. Gao, J. Chao et al., Application of radial basis functions neural networks in spectral functions. *Phys. Rev. D* **104**, 076011 (2021). <https://doi.org/10.1103/PhysRevD.104.076011>
124. S. Shi, K. Zhou, J. Zhao et al., Heavy quark potential in the quark-gluon plasma: deep neural network meets lattice quantum chromodynamics. *Phys. Rev. D* **105**, 014017 (2022). <https://doi.org/10.1103/PhysRevD.105.014017>
125. R. Larsen, S. Meinel, S. Mukherjee et al., Excited bottomonia in quark-gluon plasma from lattice QCD. *Phys. Lett. B* **800**, 135119 (2020). <https://doi.org/10.1016/j.physletb.2019.135119>
126. D. Lafferty, A. Rothkopf, Improved Gauss law model and in-medium heavy quarkonium at finite density and velocity. *Phys. Rev. D* **101**, 056010 (2020). <https://doi.org/10.1103/PhysRevD.101.056010>
127. D.S. Broomhead, D. Lowe, *Radial basis functions, multi-variable functional interpolation and adaptive networks* (Tech. rep, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988)
128. F. Schwenker, H.A. Kestler, G. Palm, Three learning phases for radial-basis-function networks. *Neural Netw.* **14**, 439–458 (2001)
129. L. Beheim, A. Zitouni, F. Belloir, et al., New rbf neural network classifier with optimized hidden neurons number. *WSEAS Trans. Syst.* 467–472 (2004)
130. J. Wang, G. Liu, A point interpolation meshless method based on radial basis functions. *Int. J. Numer. Meth. Eng.* **54**, 1623–1648 (2002)
131. J.C. Carr, W.R. Fright, R.K. Beatson, Surface interpolation with radial basis functions for medical imaging. *IEEE Trans. Med. Imaging* **16**, 96–107 (1997)
132. W. Chen, X. Han, G. Li, et al., in *Deep rbfnet: Point Cloud Feature Learning using Radial Basis Functions* (2018) [arXiv preprint arXiv:1812.04302](https://arxiv.org/abs/1812.04302)
133. H. Yoon, J.H. Sim, M.J. Han, Analytic continuation via domain knowledge free machine learning. *Phys. Rev. B* **98**, 245101 (2018). <https://doi.org/10.1103/PhysRevB.98.245101>
134. R. Fournier, L. Wang, O.V. Yazyev et al., Artificial neural network approach to the analytic continuation problem. *Phys. Rev. Lett.* **124**, 056401 (2020). <https://doi.org/10.1103/PhysRevLett.124.056401>
135. K. Zhou, N. Xu, Z. Xu et al., Medium effects on charmonium production at ultrarelativistic energies available at the CERN Large Hadron Collider. *Phys. Rev. C* **89**, 054911 (2014). <https://doi.org/10.1103/PhysRevC.89.054911>
136. J. Zhao, K. Zhou, S. Chen et al., Heavy flavors under extreme conditions in high energy nuclear collisions. *Prog. Part. Nucl. Phys.* **114**, 103801 (2020). <https://doi.org/10.1016/j.ppnp.2020.103801>
137. M. Laine, O. Philipsen, P. Romatschke et al., Real-time static potential in hot QCD. *JHEP* **03**, 054 (2007). <https://doi.org/10.1088/1126-6708/2007/03/054>
138. A. Beraudo, J.P. Blaizot, C. Ratti, Real and imaginary-time Q anti-Q correlators in a thermal medium. *Nucl. Phys. A* **806**,

- 312–338 (2008). <https://doi.org/10.1016/j.nuclphysa.2008.03.001>
139. N. Brambilla, J. Ghiglieri, A. Vairo et al., Static quark-antiquark pairs at finite temperature. *Phys. Rev. D* **78**, 014017 (2008). <https://doi.org/10.1103/PhysRevD.78.014017>
  140. N. Brambilla, M.A. Escobedo, J. Ghiglieri et al., Heavy Quarkonium in a weakly-coupled quark-gluon plasma below the melting temperature. *JHEP* **09**, 038 (2010). [https://doi.org/10.1007/JHEP09\(2010\)038](https://doi.org/10.1007/JHEP09(2010)038)
  141. B. Chen, K. Zhou, P. Zhuang, Mean field effect on  $J/\psi$  production in heavy ion collisions. *Phys. Rev. C* **86**, 034906 (2012). <https://doi.org/10.1103/PhysRevC.86.034906>
  142. F.P. Li, H.L. Lü, L.G. Pang, et al., in *Deep-Learning Quasi-particle Masses from QCD Equation of State* (2022) [arXiv:2211.07994](https://arxiv.org/abs/2211.07994)
  143. R. Baier, Y.L. Dokshitzer, A.H. Mueller et al., Radiative energy loss of high-energy quarks and gluons in a finite volume quark-gluon plasma. *Nucl. Phys. B* **483**, 291–320 (1997). [https://doi.org/10.1016/S0550-3213\(96\)00553-6](https://doi.org/10.1016/S0550-3213(96)00553-6)
  144. R. Baier, Y.L. Dokshitzer, A.H. Mueller et al., Radiative energy loss and  $p(T)$  broadening of high-energy partons in nuclei. *Nucl. Phys. B* **484**, 265–282 (1997). [https://doi.org/10.1016/S0550-3213\(96\)00581-0](https://doi.org/10.1016/S0550-3213(96)00581-0)
  145. M. Gyulassy, X.N. Wang, Multiple collisions and induced gluon Bremsstrahlung in QCD. *Nucl. Phys. B* **420**, 583–614 (1994). [https://doi.org/10.1016/0550-3213\(94\)90079-5](https://doi.org/10.1016/0550-3213(94)90079-5)
  146. X.F. Guo, X.N. Wang, Multiple scattering, parton energy loss and modified fragmentation functions in deeply inelastic  $eA$  scattering. *Phys. Rev. Lett.* **85**, 3591–3594 (2000). <https://doi.org/10.1103/PhysRevLett.85.3591>
  147. U.A. Wiedemann, Gluon radiation off hard quarks in a nuclear environment: opacity expansion. *Nucl. Phys. B* **588**, 303–344 (2000). [https://doi.org/10.1016/S0550-3213\(00\)00457-0](https://doi.org/10.1016/S0550-3213(00)00457-0)
  148. Y. Xu, J.E. Bernhard, S.A. Bass et al., Data-driven analysis for the temperature and momentum dependence of the heavy-quark diffusion coefficient in relativistic heavy-ion collisions. *Phys. Rev. C* **97**, 014907 (2018). <https://doi.org/10.1103/PhysRevC.97.014907>
  149. Y. He, L.G. Pang, X.N. Wang, Bayesian extraction of jet energy loss distributions in heavy-ion collisions. *Phys. Rev. Lett.* **122**, 252302 (2019). <https://doi.org/10.1103/PhysRevLett.122.252302>
  150. R. Soltz, Bayesian extraction of  $\hat{q}$  with multi-stage jet evolution approach. *PoS* **2018**, 048 (2019). <https://doi.org/10.22323/1.345.0048>
  151. M. Xie, W. Ke, H. Zhang, et al., in *Information Field Based Global Bayesian Inference of the Jet Transport Coefficient* (2022) [arXiv:2206.01340](https://arxiv.org/abs/2206.01340)
  152. M. Xie, W. Ke, H. Zhang, et al., in *Global Constraint on the Jet Transport Coefficient from Single Hadron, Dihadron and  $n$ -Hadron Spectra in High-Energy Heavy-Ion Collisions* (2022) [arXiv:2208.14419](https://arxiv.org/abs/2208.14419)
  153. M. Feickert, B. Nachman, in *A Living Review of Machine Learning for Particle Physics* (2021) [arXiv:2102.02770](https://arxiv.org/abs/2102.02770)
  154. K.T. Yi-Lun, D. Pablos, Applications of deep learning in jet quenching. *Sci. China Phys. Mech. Astron.* **52**, 252017 (2022). <https://doi.org/10.1360/SSPMA-2022-0046>
  155. Y.L. Du, D. Pablos, K. Tywoniuk, Deep learning jet modifications in heavy-ion collisions. *JHEP* **21**, 206 (2020). [https://doi.org/10.1007/JHEP03\(2021\)206](https://doi.org/10.1007/JHEP03(2021)206)
  156. Y.L. Du, D. Pablos, K. Tywoniuk, Jet tomography in heavy-ion collisions with deep learning. *Phys. Rev. Lett.* **128**, 012301 (2022). <https://doi.org/10.1103/PhysRevLett.128.012301>
  157. Z. Yang, Y. He, W. Chen, et al., in *Deep Learning Assisted Jet Tomography for the Study of Mach Cones in QGP* (2022) [arXiv:2206.02393](https://arxiv.org/abs/2206.02393)
  158. Y. He, T. Luo, X.N. Wang, et al., Linear Boltzmann transport for jet propagation in the quark-gluon plasma: elastic processes and medium recoil. *Phys. Rev. C* **91**, 054908 (2015). [Erratum: *Phys. Rev. C* **97**, 019902 (2018)]. <https://doi.org/10.1103/PhysRevC.91.054908>
  159. S. Cao et al., Multistage Monte-Carlo simulation of jet modification in a static medium. *Phys. Rev. C* **96**, 024909 (2017). <https://doi.org/10.1103/PhysRevC.96.024909>
  160. F.L. Liu, W.J. Xing, X.Y. Wu et al., QLBT: a linear Boltzmann transport model for heavy quarks in a quark-gluon plasma of quasi-particles. *Eur. Phys. J. C* **82**, 350 (2022). <https://doi.org/10.1140/epjc/s10052-022-10308-x>
  161. F.G. Gardim, F. Grassi, M. Luzum et al., Breaking of factorization of two-particle correlations in hydrodynamics. *Phys. Rev. C* **87**, 031901 (2013). <https://doi.org/10.1103/PhysRevC.87.031901>
  162. S.A. Voloshin, A.M. Poskanzer, R. Snellings, Collective phenomena in non-central nuclear collisions. *Landolt-Bornstein* **23**, 293–333 (2010). [https://doi.org/10.1007/978-3-642-01539-7\\_10](https://doi.org/10.1007/978-3-642-01539-7_10)
  163. R. Snellings, Elliptic flow: a brief review. *New J. Phys.* **13**, 055008 (2011). <https://doi.org/10.1088/1367-2630/13/5/055008>
  164. J. Jia, Event-shape fluctuations and flow correlations in ultra-relativistic heavy-ion collisions. *J. Phys. G* **41**, 124003 (2014). <https://doi.org/10.1088/0954-3899/41/12/124003>
  165. Z. Liu, W. Zhao, H. Song, Principal component analysis of collective flow in relativistic heavy-ion collisions. *Eur. Phys. J. C* **79**, 870 (2019). <https://doi.org/10.1140/epjc/s10052-019-7379-y>
  166. R.S. Bhalerao, J.Y. Ollitrault, S. Pal et al., Principal component analysis of event-by-event fluctuations. *Phys. Rev. Lett.* **114**, 152301 (2015). <https://doi.org/10.1103/PhysRevLett.114.152301>
  167. A. Mazeliauskas, D. Teaney, Subleading harmonic flows in hydrodynamic simulations of heavy ion collisions. *Phys. Rev. C* **91**, 044902 (2015). <https://doi.org/10.1103/PhysRevC.91.044902>
  168. A. Mazeliauskas, D. Teaney, Fluctuations of harmonic and radial flow in heavy ion collisions with principal components. *Phys. Rev. C* **93**, 024913 (2016). <https://doi.org/10.1103/PhysRevC.93.024913>
  169. P. Bozek, Principal component analysis of the nonlinear coupling of harmonic modes in heavy-ion collisions. *Phys. Rev. C* **97**, 034905 (2018). <https://doi.org/10.1103/PhysRevC.97.034905>
  170. A.M. Sirunyan et al., Principal-component analysis of two-particle azimuthal correlations in PbPb and  $p$  Pb collisions at CMS. *Phys. Rev. C* **96**, 064902 (2017). <https://doi.org/10.1103/PhysRevC.96.064902>
  171. Z. Liu, A. Behera, H. Song et al., Robustness of principal component analysis of harmonic flow in heavy ion collisions. *Phys. Rev. C* **102**, 024911 (2020). <https://doi.org/10.1103/PhysRevC.102.024911>
  172. Y.S. Zhao, L. Wang, K. Zhou et al., Detecting the chiral magnetic effect via deep learning. *Phys. Rev. C* **106**, L051901 (2022). <https://doi.org/10.1103/PhysRevC.106.L051901>
  173. K. Lee, J. Mulligan, M. Płoskoń et al., Machine learning-based jet and event classification at the Electron-Ion Collider with applications to hadron structure and spin physics. *J. High Energy Phys.* **3**, 1–35 (2023)
  174. Z. Hao, R. Kansal, J. Duarte, et al., in *Lorentz Group Equivariant Autoencoders* (2022) [arXiv:2212.07347](https://arxiv.org/abs/2212.07347)