Machine Learning Simulation of Collective Flow and Nuclear Stopping Power in Heavy Ion Collisions

Authors: Wei Guojun, Wang Yongjia, Li Qingfeng, Liu Fuhu

Date: 2025-03-28T00:00:00+00:00

Abstract

Heavy ion collision experiments combined with transport model simulations constitute one of the important tools for extracting information on nuclear matter properties. Bayesian analysis is a statistical method capable of extracting information from comparisons between experimental data and theoretical calculations, and is therefore widely employed. This process typically requires sampling in parameter space using Markov Chain Monte Carlo methods, followed by transport model simulation calculations. However, due to the complexity and extreme computational cost of transport model calculations, a transport model emulator must be trained via machine learning algorithms to accelerate this process. In this work, three machine learning algorithms—Gaussian Process, Multi-task Neural Network, and Random Forest—are employed to train an emulator for the Ultra-relativistic Quantum Molecular Dynamics (UrQMD) transport model. Within the prior distribution ranges of three parameters related to nuclear matter properties, 150 parameter sets were selected as the training set. The UrQMD transport model was used to simulate gold-gold collisions at an incident energy of 0.25 GeV per nucleon under these parameters, and observables including the directed flow, elliptic flow of free protons, and nuclear stopping power were extracted from final-state particle information. Additionally, 20 randomly selected parameter sets were used as a test set to evaluate the emulator' s performance. The results demonstrate that the coefficients of determination (R^2) for predictions on the test set are 0.95, 0.93, and 0.85 for the Gaussian Process, Multi-task Neural Network, and Random Forest algorithms, respectively, indicating that both Gaussian Process and Multi-task Neural Network exhibit high accuracy in emulating observables calculated by transport models and can effectively accelerate the transport model calculation process.

Full Text

Preamble

Vol. XX, No. X, XXX 20XX
NUCLEAR TECHNIQUES

Machine Learning Simulates Collective Flow and Nuclear Stopping in Heavy-ion Collisions at Intermediate Energies

Guojun Wei[1],[2], Yongjia Wang[2], *Qingfeng Li*[2], Fuhu Liu[1]
[1] Institute of Theoretical Physics, Shanxi University, Taiyuan 030006, China
[2] School of Science, Huzhou University, Huzhou 313000, China

Abstract

[**Background**] The nuclear equation of state (EoS) delineates the thermodynamic relationship between nucleon energy and nuclear matter density, temperature, and isospin asymmetry. This relationship is crucial for validating existing nuclear theoretical models, investigating the nature of nuclear forces, and understanding the structure of compact stars, neutron star mergers, and related astrophysical processes. Heavy-ion collision experiments combined with transport models represent a critical approach for exploring the high-density behavior of the EoS. With the rapid advancement of a new generation of high-current heavy-ion accelerators and the development of efficient detection technologies, the variety, volume, and precision of data generated from heavy-ion collision experiments have seen substantial enhancements. The effective utilization and analysis of these experimental datasets to extract crucial information about the EoS have emerged as one of the central challenges in contemporary heavy-ion physics research.

Bayesian analysis is a statistical method that can extract reliable physical information by comparing experimental data with theoretical calculations while quantifying parameter uncertainties, and thus has attracted widespread attention. In the task of determining EoS parameter ranges through Bayesian inference, Monte Carlo sampling methods are used to extract observables from final-state particle information simulated by transport models for each set of EoS parameters. Since the calculation process of transport models is complex, this step consumes considerable time, significantly hindering the efficiency of data generation and limiting the ability to explore the full parameter space.

[**Purpose**] To address this challenge, there is a pressing need for a more efficient approach to simulate transport models, particularly one that leverages modern computational techniques to accelerate the process.

[**Methods**] Here, we propose a machine learning-based approach to develop a transport model emulator that can significantly reduce calculation time. We evaluate three machine learning algorithms—Gaussian processes, multi-task neural networks, and random forests—to train emulators based

on the ultra-relativistic quantum molecular dynamics (UrQMD) transport model. The selected observables are protons' directed flow, elliptical flow, and nuclear stopping extracted from the final state of Au+Au collisions at $E_{\rm lab} = 0.25$ GeV/nucleon across different EoS parameters (incompressibility K_0, effective mass m^*, and in-medium correction factor F of nucleon-nucleon elastic cross-section). We run the UrQMD model with 150 parameter sets: $K_0 = 180, 220, 260, 300, 340, 380$ MeV, $m^*/m = 0.6, 0.7, 0.8, 0.9, 0.95$, and $F = 0.6, 0.7, 0.8, 0.9, 1.0$. For each case, 2×10^5 events with reduced impact parameter $b_0 < 0.45$ are simulated to ensure negligible statistical errors. The results from these 150 parameter sets are fed to the three machine learning algorithms to train the emulators. Additionally, 20 parameter sets with randomly chosen K_0, m^*, and F are run, and the resulting observables are used to test emulator performance.

[**Results**] The results obtained from Gaussian processes and multi-task neural networks align well with those calculated by the UrQMD model, indicating that these two emulators achieve high accuracy and can be used in Bayesian analysis. However, when predicting with reduced impact parameter $b_0 < 0.25$, some data points predicted by random forest show large errors, indicating that random forest performs relatively poorly as a transport model emulator for predicting observables. To further compare the prediction effectiveness of the three transport model emulators, we select the coefficient of determination R^2 as the evaluation metric. The R^2 values for Gaussian process, multi-task neural network, and random forest on the test set are 0.95, 0.93, and 0.85, respectively. These results show that both Gaussian process and multi-task neural network demonstrate high accuracy when simulating UrQMD model data and can effectively accelerate the calculation process. However, for complex tasks with numerous parameters and observables, the efficiency and accuracy of Gaussian process emulators may suffer. Thus, relying solely on Gaussian processes may not suffice. In such cases, multi-task neural networks show greater adaptability, better handling complex datasets and effectively learning information within parameter spaces.

[**Conclusions**] In summary, Gaussian processes generally perform well as transport model emulators in Bayesian frameworks, particularly for moderate-sized datasets, while multi-task neural networks may be a more ideal choice for complex tasks with more parameters and observables. In practical applications, the most suitable emulator should be selected according to specific task requirements and data characteristics.

Keywords: Heavy-ion collisions, Nuclear equation of state, Transport model, Machine learning, Emulator

1. UrQMD Transport Model and Heavy-ion Collision Data

1.1 UrQMD Transport Model

The Ultra-relativistic Quantum Molecular Dynamics (UrQMD) model is a many-body microscopic transport model that primarily includes projectile-target initialization, mean field, collision term, and fragment construction components. In this model, each nucleon is represented by a Gaussian wave packet with a definite width [30-32]:

$$\phi(\mathbf{x};\mathbf{p}) = \left(\frac{2\alpha}{\pi}\right)^{3/4} \exp\left[-\alpha(\mathbf{x}-\mathbf{r}(t))^2 + \frac{i\mathbf{p}(t)\cdot\mathbf{x}}{\hbar}\right]$$

Nucleon coordinates and momenta are sampled randomly under certain constraints. The coordinate range extends from 0 to the nuclear radius R, while the momentum range extends from 0 to the local Fermi momentum $p_F(\mathbf{r}) = \hbar(3\pi^2\rho(\mathbf{r})/2)^{1/3}$. Additionally, distinctions between protons and neutrons and other constraints are considered.

After initialization, heavy-ion collisions evolve under the combined action of the mean field and collision term. The evolution of the coordinate $\mathbf{r}_i$ of the i-th nucleon wave packet center follows:

$$\dot{\mathbf{r}}_i = \frac{\partial H}{\partial \mathbf{p}_i}, \quad \dot{\mathbf{p}}_i = -\frac{\partial H}{\partial \mathbf{r}_i}$$

where H is the total Hamiltonian of the system, consisting of kinetic and potential energy. The nucleon potential energy $U_i = \partial V_{\text{pot}}/\partial\rho_i$ is calculated from the potential energy density [33-35], which can be written as:

$$V_{\text{pot}} = \frac{A}{2}\frac{\rho^2}{\rho_0} + \frac{B}{\sigma+1}\frac{\rho^{\sigma+1}}{\rho_0^\sigma} + \frac{C_{\text{surf}}}{2\rho_0}(\nabla\rho)^2 + \frac{C_\tau}{2\rho_0}(\nabla\rho)^2 + \frac{C_{\text{sym}}}{2\rho_0}\rho\delta^2\left[1-\frac{2}{3}\left(\frac{\rho}{\rho_0}\right)^\gamma\right] + V_{\text{mom}}$$

The terms include two-body and three-body interaction components. The surface term and surface asymmetry term are $E_{\text{surf}} = \frac{C_{\text{surf}}}{2\rho_0}(\nabla\rho)^2$ and $E_{\text{surf}}^{\text{asy}} = \frac{C_\tau}{2\rho_0}(\nabla\rho)^2$, respectively. $E_{\text{sym}} = \frac{C_{\text{sym}}}{2\rho_0}\rho\delta^2\left[1-\frac{2}{3}\left(\frac{\rho}{\rho_0}\right)^\gamma\right]$ is the nuclear symmetry energy term. In this work, we use the MSL0 parameter set from Skyrme interactions with parameters $A = 36.4$ MeV, $B = 19.5$ MeV $\cdot$ fm^2, $C_{\text{surf}} = -11.3$ MeV $\cdot$ fm^2, $C_\tau = -15.1$ MeV, $C = -3.5$ MeV, and $\sigma = 1.2359$. The symmetry energy slope parameter L is 60 MeV.

The momentum-dependent term contribution can be written as:

$$V_{\text{mom}} = \frac{1}{2\rho_0}\sum_{i,j}\int\int f(\mathbf{r}_i,\mathbf{p}_i)f(\mathbf{r}_j,\mathbf{p}_j)\mathbf{v}_{ij}\cdot\mathbf{v}_{ij}\left[1+\left(\frac{\rho}{\rho_0}\right)^\gamma\right]d_i^{3r}d_i^{3p}$$

For zero-temperature isospin-symmetric nuclear matter, the binding energy can be expressed as:

$$E_0(\rho) = \frac{3}{5}E_F(\rho) + \frac{A}{2}\frac{\rho}{\rho_0} + \frac{B}{\sigma+1}\left(\frac{\rho}{\rho_0}\right)^{\sigma}$$

The momentum-dependent contribution can be written as:

$$E_{\mathrm{mom}}(\rho) = \frac{1}{5}E_F(\rho)\left[1 + \left(\frac{\rho}{\rho_0}\right)^{\gamma}\right]$$

1.2 Heavy-ion Collision Data

Simulating heavy-ion collisions via the UrQMD transport model and extracting observables from particle momentum information for comparison with experimental data represents an important approach for studying heavy-ion reactions. Collective flow is a crucial observable that describes the collective motion of particles produced in heavy-ion collisions. Initially, particles have momentum only along the z-axis (with small Fermi momentum components in the x and y directions). As the reaction progresses, random collisions between particles and expansion of hot dense matter lead to momentum components along the x and y axes comparable to the z direction, forming collective motion within the reaction plane known as collective flow [36-40]. This phenomenon is closely related to the spatiotemporal evolution of collision dynamics and the nuclear matter equation of state, serving as an important tool for exploring nuclear matter properties and holding significance for many topics across a wide range of heavy-ion collision energies.

Directed flow and elliptic flow are the most commonly used flow observables. Directed flow reflects the emission effect of particles in the reaction plane and can be obtained from the Fourier expansion of particle azimuthal distribution:

$$v_1 = \langle \cos(\phi - \Psi_{\mathrm{RP}}) \rangle$$

where ϕ is the particle azimuthal angle and Ψ_{RP} is the reaction plane angle. Elliptic flow reflects the squeeze-out effect of particles perpendicular to the reaction plane and is defined as:

$$v_2 = \langle \cos[2(\phi - \Psi_{\mathrm{RP}})] \rangle$$

The distributions of v_1 and v_2 with reduced rapidity y_0 can typically be fitted with polynomial functions. We are particularly interested in the slope of directed flow at mid-rapidity $dv_1/dy_0|_{y_0=0}$ and the value of elliptic flow at $y_0 = 0$.

In addition to collective flow, nuclear stopping power is another important observable in heavy-ion collisions, describing the efficiency of converting longitudinal beam energy into transverse motion. A common observable representing nuclear stopping capability is defined as the ratio of variance in particle velocity distribution in the transverse direction to that in the longitudinal direction [41-44]:

$$\mathrm{var}_{xz} = \frac{\sigma_x^2}{\sigma_z^2}$$

where σ_x^2 and σ_z^2 are the variances of rapidity distributions in the x and z directions, respectively, and N is the number of particles in the corresponding rapidity interval.

2. Machine Learning Algorithms

2.1 Gaussian Process

In machine learning, Gaussian process is a supervised modeling approach that effectively captures uncertainty and complex patterns in data by defining probability distributions over input spaces. Both regression and classification predictions provide confidence intervals in the form of Gaussian distribution standard deviations. The fundamental principle of Gaussian process regression involves defining a mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ to calculate the covariance matrix. A commonly used covariance function is the Radial Basis Function (RBF) kernel, which defines the correlation between any two input points $\mathbf{x}$ and $\mathbf{x}'$ with the analytical form:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$$

where σ_f and ℓ are hyperparameters representing amplitude and characteristic length scale.

During training, given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the model is constructed by estimating hyperparameters using the mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$. In the prediction phase, for new input points $\mathbf{x}^*$, Gaussian process predicts the output, yielding predictive mean μ^* and uncertainty (variance) σ^{*2}:

$$\mu^* = \mathbf{k}(\mathbf{x}^*, \mathbf{X})\mathbf{K}^{-1}\mathbf{y}$$
$$\sigma^{*2} = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, \mathbf{X})\mathbf{K}^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}^*)$$

where $\mathbf{y}$ is the observation vector, $\mathbf{K}$ is the covariance matrix of training data, and $\mathbf{k}(\mathbf{x}^*, \mathbf{X})$ is the covariance vector between new input points and existing data points [45-47]. Gaussian process performance heavily depends on hyperparameters in the covariance function, typically learned by maximizing the marginal

likelihood function using optimization methods like gradient descent to effectively find optimal hyperparameter combinations. Gaussian processes can be implemented using libraries such as GPy or scikit-learn.

Gaussian processes offer unique advantages in many scenarios. First, by basing predictions on the covariance matrix from training data, they can make reasonable inferences even with limited data, generating high-quality predictions when samples are scarce. Second, when handling complex unknown nonlinear relationships, Gaussian processes can flexibly adapt to underlying data structures through appropriate covariance function selection while providing confidence levels for each prediction point. Additionally, Gaussian process predictions include not only point estimates but also variance estimates, providing clear guidance for understanding model behavior.

2.2 Neural Networks

Neural networks, as core technology in artificial intelligence and machine learning, simulate biological neuron mechanisms and possess powerful pattern recognition and nonlinear mapping capabilities. They consist of interconnected nodes (or "neurons") organized into network layers through which data propagates and transforms to produce final outputs [48-49].

The basic structure includes an input layer, hidden layers, and an output layer. The input layer receives external data, with each node corresponding to one feature; hidden layers, located between input and output layers, can be single or multiple, with their number and neuron count determining network capacity and expressive power; the output layer generates final predictions, with its structure depending on the specific task (classification or regression). Each neuron connects to all neurons in the previous layer, receiving their input values to compute a weighted sum $z = \sum_{i=1}^{n} w_i x_i + b$, which then passes through a nonlinear activation function $y = f(z)$ to produce the neuron' s output. Activation functions introduce nonlinearity, enabling neural networks to handle complex nonlinear relationships. Common activation functions include Sigmoid, Tanh, and ReLU.

Neural network training aims to adjust weights and biases to find optimal parameters θ that minimize error between output results and target values. Training typically relies on backpropagation, which uses the chain rule to compute gradients of the loss function relative to parameters in each layer and propagates errors backward, adjusting network weights layer by layer using gradient descent. The loss function measures differences between neural network outputs (predictions) and actual target values [50-53]. Common gradient descent algorithms include SGD, Adagrad, RMSProp, and Adam. Typical loss functions are mean squared error (MSE) for regression tasks and cross-entropy loss for classification tasks.

With increasing computational power and large-scale datasets, deep neural networks (DNNs) have gradually become mainstream. Deep networks significantly

enhance model expressive power by increasing hidden layer numbers. Common neural network types include Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN), all of which can be implemented using libraries such as PyTorch, TensorFlow, Keras, and sklearn.

2.3 Decision Trees and Random Forest

Decision trees are classification and regression models based on tree structures. The main idea is to classify or regress data through a series of decision rules (splitting datasets according to feature values). Decision trees recursively partition data by selecting optimal splitting features at each node until the dataset is divided into minimal subsets or preset stopping conditions are reached.

During construction, information gain is typically used to select optimal splitting features. Information gain is based on information entropy:

$$H(D) = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

where K is the number of categories and p_k is the probability of category k in dataset D. Information gain is calculated as:

$$\text{Gain}(D, A) = H(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} H(D_v)$$

where V is the number of values for feature A and D_v is the sample subset when feature A takes value v. Decision trees offer advantages of easy interpretation and strong explainability, making them suitable for data with clear decision rules. However, they are prone to overfitting, especially when tree depth is excessive. To address this, pruning techniques are often employed for optimization [54].

Random forest is an ensemble model consisting of multiple decision trees, with its core idea being "ensemble learning" —combining predictions from multiple decision trees to improve overall performance. Random forest uses bootstrap sampling to randomly generate multiple different training sets from the original dataset, building one decision tree per sampled subset. This means each decision tree trains on a different subset of the original data. During tree construction, only a random subset of features is selected for splitting at each node, increasing model diversity and reducing correlation between trees. For prediction, random forest determines final classification results through majority voting across multiple decision trees for classification tasks, or averages predictions from multiple trees for regression tasks [55-56].

3. Results and Analysis

This study selects three parameters related to the nuclear equation of state: incompressibility coefficient K_0, nucleon effective mass m^*, and nucleon-nucleon collision cross-section medium correction factor F, with prior distributions shown in Table 1 . The observable data used are free proton v_1, v_2, and var_{xz} from Au+Au collisions at incident energy 0.25 GeV/nucleon, as detailed in Table 2 .

The Gaussian process emulator utilizes known simulation or experimental data to fit a Gaussian process model capable of predicting new input points, commonly applied for model data generation and sampling in Bayesian analysis. We select 150 parameter sets as training data and 20 parameter sets as test data for the Gaussian process transport model emulator. Through Gaussian simulation and principal component analysis, UrQMD model predictions are effectively interpolated across parameter sets. In the implementation, we use a combination of ConstantKernel and Radial Basis Function (RBF) as the kernel function, both with initial length scale of 1 and value range between 10^{-4} and 1. To find optimal hyperparameter combinations, we define a hyperparameter grid and employ grid search with 5-fold cross-validation to evaluate model performance and identify the best hyperparameter combination.

Figure 1 [Figure 1: see original paper] shows the prediction results of the trained Gaussian process emulator on the test set. In the figure, the red line represents the baseline from UrQMD model calculations, while black stars show emulator predictions—the closer the black stars are to the red line, the higher the emulator accuracy. The results show that almost all predicted data points cluster near the baseline with deviations within error bounds, indicating high emulator accuracy and its ability to replace transport models for accelerated data generation. To further verify emulator reliability, we calculate Pearson correlation coefficients between observables. Figure 2 [Figure 2: see original paper] displays the Pearson correlation coefficient matrix between model-calculated values and emulator-predicted values, showing consistent correlations that demonstrate the emulator has indeed learned the intrinsic physical relationships within observable data.

We also employ multi-task neural networks and random forest algorithms as transport model emulators. The multi-task neural network has one hidden layer with 50 neurons, uses ReLU activation function, and updates parameters via the Adam optimizer. The random forest algorithm generates 100 decision trees with no limit on maximum depth, minimum samples per leaf of 2, and minimum samples to split an internal node of 3. Figure 3 [Figure 3: see original paper] shows prediction results from these trained emulators. Multi-task neural network predictions generally fall within error bounds, while random forest shows several data points with large deviations from the baseline when predicting v_{11} and v_{20} at reduced impact parameter $b_0 < 0.25$, indicating relatively poor performance as a transport model emulator.

To further compare the three emulators' prediction effectiveness, we select the

coefficient of determination R^2 as the evaluation metric, with results shown in Table 3 . Random forest demonstrates excellent performance on the training set with R^2 up to 0.98, indicating strong fitting capability. However, on the test set, R^2 is only 0.85, suggesting potential overfitting due to sparse and limited training data where random forest performs relatively poorly. In contrast, multi-task neural network and Gaussian process show more stable predictions with identical R^2 values of 0.93 and 0.95 on both training and test sets, respectively, demonstrating effective automatic learning and information extraction from observational data. However, multi-task neural network training time is longer, which may limit its application, while Gaussian process computational complexity is higher, requiring substantial resources for large-scale datasets and being more suitable for small-scale datasets.

In summary, the three transport model emulators each have advantages and disadvantages: random forest, despite strong fitting capability, carries overfitting risk; multi-task neural networks can capture complex nonlinear relationships but require longer training; Gaussian processes have strong generalization ability but consume more computational resources. In practice, emulator selection should be based on specific task requirements and data characteristics. Generally, Gaussian processes are suitable for most tasks with fewer parameters and observables, while multi-task neural networks are more effective for large-scale data scenarios.

After training the transport model emulators, we conducted rigorous Bayesian analysis to extract posterior likelihood distributions of parameters characterizing nuclear matter properties. Results show that effective mass m^* and medium correction factor F can be constrained to Gaussian distributions, while incompressibility K_0 remains difficult to constrain.

Transport model emulators combined with experimental data can constrain nuclear equation of state parameters through Bayesian analysis while considering multiple experimental observables and providing parameter ranges. Future work will incorporate the latest heavy-ion collision experimental data and transport model calculations to constrain parameters describing the nuclear equation of state and nucleon-nucleon scattering cross-sections in nuclear medium, with related research currently underway.

Acknowledgments

The authors thank TMEP group members for discussions, C3S2 Computing Center of Huzhou University for support, and acknowledge funding from the National Natural Science Foundation of China (Grant No. 12335008), National Key R&D Program (Grant No. 2023YFA1606402), National Key Laboratory of Plasma Physics Fund (Grant No. 6142A04230203), Zhejiang Provincial Education Department Research Project (Grant No. Y202353782), and Huzhou Natural Science Foundation (Grant No. 2024YZ28).

References

1. Sorensen A, Agarwal K, Brown K W, et al. Dense nuclear matter equation of state from heavy-ion collisions[J]. Progress in Particle and Nuclear Physics, 2024, 134: 104080.
2. Wang J M, Deng X G, Xie W J, et al. Bayesian inference of nuclear incompressibility from proton elliptic flow in central Au+Au collisions at 400 MeV/nucleon[J]. arxiv: preprint arxiv:2406.07051 (2024).
3. Wang Y J, Li Q F, Leifels Y, et al. Study of the nuclear symmetry energy from the rapidity-dependent elliptic flow in heavy-ion collisions around 1 GeV/nucleon regime[J]. Physics Letters B, 2020, 802: 135249.
4. Xu J. Transport approaches for the description of intermediate-energy heavy-ion collisions[J]. Progress in Particle and Nuclear Physics, 2019, 106: 312-359.
5. Li B A, Ko C M, Bauer W. Isospin physics in heavy-ion collisions at intermediate energies[J]. International Journal of Modern Physics E, 1998, 7(02): 147-229.
6. Li H L, Xu H J, Zhou Y, et al. Probing the neutron skin with ultrarelativistic isobaric collisions. Physical Review Letters, 2020, 125(22): 222301.
7. Xu H J, Li H L, Zhou Y, et al. Measuring neutron skin by grazing isobaric collisions. Physical Review C, 2022, 105(1): L011901.
8. Omana K M, Steinheimer J, Zhou K, et al. Qcd equation of state of dense nuclear matter from a bayesian analysis of heavy-ion collision data[J]. Physical Review Letters, 2023, 131(20): 202303.
9. Tsang C Y, Kurata-Nishimura, Tsang M B, et al. Constraining nucleon effective masses with flow and stopping observables from the SπRIT experiment[J]. Physics Letters B, 2024, 853: 138661.
10. Mäntysaari H, Schenke B, Shen C, et al. Bayesian inference of the fluctuating proton shape[J]. Physics Letters B, 2022, 833: 137515.
11. Fan W, Vujanovic G, Bass S A, et al. New metric improving Bayesian calibration of a multistage approach studying hadron and inclusive jet suppression[J]. Physical Review C, 2024, 109(6): 064903.
12. Everett D, Ke W, Paquet J F, et al. Multisystem Bayesian constraints on the transport coefficients of QCD matter[J]. Physical Review C, 2021, 103(5): 054904.
13. Zhou J, Xu J. Bayesian inference of neutron-skin thickness and neutron-star observables based on effective nuclear interactions. Science China Physics, Mechanics & Astronomy, 2024, 67(8): 282011.
14. Zhang Z, Chen L W. Bayesian inference of the symmetry energy and the neutron skin in 48Ca and 208Pb from CREX and PREX-2[J]. Physical Review C, 2023, 108(2): 024317.
15. Cox N, Grundler X, Li B A. Neural network emulation of flow in heavy-ion collisions at intermediate energies[J]. Physical Review C, 2024, 110(4): 044604.
16. Seeger, Matthias. Gaussian processes for machine learning[J]. International journal of neural systems, 2004, 14(02): 69-106.

17. Boehnlein A, Diefenthaler M, Sato N, et al. Colloquium: Machine learning in nuclear physics[J]. Reviews of modern physics, 2022 94(3): 031003.
18. He W B, Li Q F, Ma Y G, et al. Machine learning in nuclear physics at low and intermediate energies[J]. Science China Physics, Mechanics & Astronomy, 2023 66(8): 282001.
19. Pang L G. Studying high-energy nuclear physics with machine learning[J]. International Journal of Modern Physics E, 2024, 33(06): 2430009.
20. Li Fupeng, Pang Longgang, Wang Xinnian. Application of machine learning to the study of QCD transition in heavy ion collisions[J]. Nuclear Techniques, 2023, 46(04): 040014-040014.
21. Gao Zepeng, Li Qingfeng. Studies on several problems in nuclear physics by using machine learning[J]. Nuclear Techniques, 2023, 46(08): 95-102.
22. Zhang Yingxun, Wang Fangyuan, Li Li, et al. Study of Physics Problem in Low-intermediate Energy Heavy Ion Collision with Machine Learning Algorithm[J]. Atomic Energy Science and Technology, 2023, 57(4): 751-761.
23. Yi Jiayi, Qiao Chunyuan, Pei Junchen, et al. Bayesian machine learning for the uncertainty evaluation of nuclear fission product yields[J]. SCIENTIA SINICA Physica, Mechanica & Astronomica, 2022, 52(5): 252013.
24. Han Shuochong, Shang Xinle, Zuo Wei. Taming the in-medium nucleon-nucleon cross section with a deep neural network method[J]. SCIENTIA SINICA Physica, Mechanica & Astronomica, 2022, 52(5): 252015.
25. Li F P, Wang Y J, Gao Z P, et al. Application of machine learning in the determination of impact parameter in the 132Sn+124Sn system. Physical Review C, 2024, 104(3): 034608.
26. Saha A, Dan D, and Sanyal S. Machine-learning model-driven prediction of the initial geometry in heavy-ion collision experiments[J]. Physical Review C, 2024, 106(1): 014901.
27. Zhang X, Liu X Q, Huang Y, et al. Determining impact parameters of heavy-ion collisions at low-intermediate incident energies using deep learning with convolutional neural networks[J]. Physical Review C, 2024, 105(3): 034611.
28. Wang Y J, Gao Z P, Lü H L, et al. Decoding the nuclear symmetry energy event-by-event in heavy-ion collisions with machine learning[J]. Physics Letters B, 2022, 835: 137508.
29. Huang Y G, Pang L G, Luo X F, et al. Probing criticality with deep learning in relativistic heavy-ion collisions[J]. Physics Letters B, 2022, 827: 137001.
30. Wang Y J, and Li Q F. Application of microscopic transport model in the study of nuclear equation of state from heavy ion collisions at intermediate energies[J]. Frontiers of Physics, 2020, 15(4): 44302.
31. Zhang Y X, Liu Y Y, Wang Y J, et al. How Do Constraints of Nuclear Symmetry Energy Reconcile with Different Models?[J]. arxiv: preprint arxiv:2301.00212 (2022).
32. Li Q F, Wang Y J, Wang X B, et al. Rapidity distribution of protons from the potential version of UrQMD model and the traditional coalescence

afterburner. Science China Physics, Mechanics & Astronomy, 2016, 59(2): 622001.
33. Liu Y Y, Wang Y J, Cui Y, et al. Insights into the pion production mechanism and the symmetry energy at high density[J]. Physical Review C, 2021, 103(1): 014616.
34. Li P C, Wang Y J, Li Q F, et al. Effects of the in-medium nucleon-nucleon cross section on collective flow and nuclear stopping in heavy-ion collisions in the Fermi-energy domain[J]. Physical Review C, 2018, 97(4): 044620.
35. Tong L Y, Li P C, Li F P, et al. Nucleon effective mass splitting and density-dependent symmetry energy effects on elliptic flow in heavy ion collisions at Elab = 0.09 ~ 1.5 GeV/nucleon. Chinese Physics C, 2020, 44(7): 074103.
36. Xu H J, Wang X B, Li H L, et al. Importance of isobar density distributions on the chiral magnetic effect search. Physical Review Letters, 2018, 121(2): 022301.
37. Xu H J, Li Z P and Song H C. High-order flow harmonics of identified hadrons in 2.76 A TeV Pb+ Pb collisions. Physical Review C, 2016, 93(6): 064905.
38. Wang Y J, Guo C C, Li Q F, et al. Determination of the nuclear incompressibility from the rapidity-dependent elliptic flow in heavy-ion collisions at beam energies 0.4 A–1.0 A GeV. Physics Letters B, 2018, 778: 207-212.
39. Xu H J, Zhao J, Wang X B, et al. Varying the chiral magnetic effect relative to flow in a single nucleus-nucleus collision. Chinese Physics C, 2018, 42(8): 084103.
40. Xu H J, Li H L, Wang X B, et al. Determine the neutron skin type by relativistic isobaric collisions. Physics Letters B, 2021, 819: 136453.
41. Li P C, Wang Y J, Li Q F, et al. Collective flow and nuclear stopping in heavy ion collisions in Fermi energy domain[J]. Nuclear Science and Techniques, 2018, 29: 1-8.
42. Bhalerao Rajeev S, Jaiswal A, PalBhalerao S. Collective flow in event-by-event partonic transport plus hydrodynamics hybrid approach[J]. Physical Review C, 2015, 92(1): 014903.
43. Zheng L, Liu L, Lin Z W, et al. Disentangling the development of collective flow in high energy proton proton collisions with a multiphase transport model[J]. The European Physical Journal C, 2024, 84(10): 1029.
44. Li P C, Wang Y J, Li Q F, et al. Accessing the in-medium effects on nucleon-nucleon elastic cross section with collective flows and nuclear stopping[J]. Physics Letters B, 2022, 828: 137019.
45. Keller J, Hebeler K, Schwenk A. Nuclear equation of state for arbitrary proton fraction and temperature based on chiral effective field theory and a Gaussian process emulator[J]. Physical Review L, 2023, 130(7): 072701.
46. Yuan Z Y, Bai D, Wang Z, et al. Reliable calculations of nuclear binding energies by the Gaussian process of machine learning[J]. Nuclear Science and Techniques, 2024, 35(6): 105.
47. Zhou S, Giulani P, Piekarewicz J, et al. Reexamining the proton-radius problem using constrained Gaussian processes[J]. Physical Review C, 2019,

99(5): 055202.
48. Shi S Z, Zhou K, Zhao J X, et al. Heavy quark potential in the quark-gluon plasma: Deep neural network meets lattice quantum chromodynamics[J]. Physical Review D, 2022, 105(1): 014017.
49. Qin F D, Luo H Y, He Z Z, et al. Counting of alpha particle tracks on imaging plate based on a convolutional neural network[J]. Nuclear Science and Techniques, 2023, 34(3): 37.
50. Pang L G, Zhou K, Wang X N. (2020). Deep learning for nuclear physics[J]. Nuclear Physics Review, 2020, 37(3), 720-726.
51. Andrieu C, De Freitas N, Doucet A, et al. An introduction to MCMC for machine learning[J]. Machine learning, 2003, 50: 5-43.
52. Baldi P, Bauer K, Eng C, et al. Jet substructure classification in high-energy physics with deep neural networks[J]. Physical Review D, 2016, 93(9): 094034.
53. Niu Z M, Fang J Y, and Niu Y F. Comparative study of radial basis function and Bayesian neural network approaches in nuclear mass predictions[J]. Physical Review C, 2019, 100(5): 054311.
54. Carnini M, and Pastore A. Trees and forests in nuclear physics[J]. Journal of Physics G: Nuclear and Particle Physics, 2020, 47(8): 082001.
55. Cai B S, Yuan C X. Random forest-based prediction of decay modes and half-lives of superheavy nuclei[J]. Nuclear Science and Techniques, 2023, 34(12), 204.
56. Reis I, Baron D, Shahaf S, et al. Probabilistic random forest: A machine learning algorithm for noisy data sets[J]. The Astronomical Journal, 2018, 157(1): 16.

Note: Figure translations are in progress. See original paper for figures.