Ultrafast, event-by-event heavy-ion simulations for next-generation experiments

Manjunath Omana Kuttan,[1,2,*] Kai Zhou,[3,1,†] Jan Steinheimer,[4,1] and Horst Stoecker[1,4,5]
[1] *Frankfurt Institute for Advanced Studies, Ruth-Moufang-Straße 1, D-60438 Frankfurt am Main, Germany*
[2] *Xidian-FIAS International Joint Research Center, Giersch Science Center, D-60438 Frankfurt am Main, Germany*
[3] *School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, People's Republic of China*
[4] *GSI Helmholtzzentrum für Schwerionenforschung GmbH, Planckstraße 1, D-64291 Darmstadt, Germany*
[5] *Institut für Theoretische Physik, Goethe Universität Frankfurt, Max-von-Laue-Straße 1, D-60438 Frankfurt am Main, Germany*

(Received 26 February 2025; revised 28 June 2025; accepted 1 August 2025; published 24 November 2025)

We present a novel deep generative framework that uses probabilistic diffusion models for ultrafast, event-by-event simulations of heavy-ion collision output. This new framework is trained on ultrarelativistic quantum molecular dynamics (UrQMD) cascade data to generate a full collision event output containing 26 distinct hadron species. The output is represented as a point cloud, where each point is defined by a particle's momentum vector and its corresponding species information. Our architecture integrates a normalizing flow-based condition generator that encodes global event features into a latent vector, and a diffusion model that synthesizes a point cloud of particles based on this condition. A detailed description of the model and an in-depth analysis of its performance is provided. The conditional point-cloud diffusion model learns to generate realistic output particles of collision events which successfully reproduce the UrQMD distributions for multiplicity, momentum, and rapidity of each hadron type. The flexible point-cloud representation of the event output preserves full event-level granularity, enabling direct application to inverse problems and parameter estimation tasks while also making it easily adaptable for accelerating any event-by-event model calculation or detector simulation.

DOI: 10.1103/wyq5-hlp5

I. INTRODUCTION

The study of strongly interacting matter under various thermodynamic conditions, governed by quantum chromodynamics (QCD), lies at the heart of relativistic heavy-ion collision experiments worldwide [1,2]. The temperature and density reached in these collisions are varied by tuning the collision energy to explore the largely conjectured QCD phase diagram. While very high collision energies ($\sqrt{s_{NN}} \gtrsim$ 200 GeV) are expected to produce systems with vanishing net-baryon densities, high-density QCD matter ($\gtrsim$2 times nuclear saturation density n_0) can be explored at intermediate collision energies ($\sqrt{s_{NN}} \approx$ 2–10 GeV). At high baryon densities, the phase structure of QCD is speculated to contain several interesting structures, such as a potential first-order phase transition from hadronic to partonic matter and a critical end point.

Lattice QCD calculations [3–8] have shown that, at vanishing net-baryon densities, hadronic matter transforms to a deconfined quark-gluon phase via a smooth crossover. However, lattice QCD calculations at finite baryon densities remain intractable due to the fermionic sign problem, necessitating reliance on effective models of QCD and advanced computational tools that are closely connected to the experiments to explore high-baryon-density QCD matter.

The experimental programs of STAR-FXT and STAR-BES at RHIC, CBM at SIS-100, HADES at SIS-18, CEE at HIAF, and MPD at NICA are dedicated to studying intermediate-energy nuclear collisions [9–16]. With unprecedented event rates and measurement precision, these experiments provide a unique opportunity to investigate the mostly unknown, high-density region of QCD phase diagram using rare observables and novel methods. The experimental efforts are complemented by various approaches for a reliable theoretical description of relativistic, moderate-energy heavy-ion collisions.

A well-motivated, microscopic, and nonequilibrium description of heavy-ion collisions at these intermediate energies is provided by well-tested transport models [17–20]. An alternative strategy is to adopt a hybrid framework, in which the hot and dense hadron matter phase is evolved via hydrodynamics while the initial and final stages, including event-by-event fluctuations, are treated through transport theory [21]. A key advantage of such a hybrid description is the explicit inclusion of various unknown types of QCD transitions or critical end points through different equations of state (EOS) modeled to drive the hydrodynamic evolution. However, important nonequilibrium phenomena such as the

*Contact author: manjunath@fias.uni-frankfurt.de
†Contact author: zhoukai@cuhk.edu.cn

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

influence of the EOS in the initial compression phase are neglected in such an approach. Recently, it was shown that a self-consistent transport description of the entire evolution with any assumed EOS can be achieved by incorporating a density-dependent potential within the QMD part of ultra-relativistic quantum molecular dynamics (UrQMD) [22,23]. Building on this concept, both momentum and particle dependence for the EOS can also be realized within the UrQMD model as demonstrated in Ref. [24]. Similar developments are also attempted in other transport model approaches [25–27].

The experimental determination of the EOS remains extremely challenging. Several observables [1,28–33] have been conjectured to be sensitive to the EOS of the hot, dense matter created in heavy-ion collisions. However, an unambiguous determination of the EOS has not been presented yet. A systematic comparison of extensive model calculations with needed experimental data for multiple observables is necessary to infer the EOS that provides the best description for experimental data. Such analyses are typically conducted using a computationally intensive Bayesian inference procedure. However, current models for event-by-event simulations are not efficient enough to calculate different observables with sufficient statistical significance for the several thousands of different parameter sets that are necessary to construct a reliable Bayesian posterior. As a result, current Bayesian inference methods rely on machine-learning-based surrogate models, which serve as fast emulators, mapping the model parameters to various observables [25,34–40]. This strategy is effective for a small number of computationally inexpensive observables such as integrated flow coefficients. However, the number of simulations necessary to generate sufficient training data quickly makes this approach infeasible if the analysis were to include a large number of observables requiring a large number of events to compute, e.g., higher-order fluctuations or multidifferential flow of various hadronic species. Hence, further advancements in both the computational efficiency of simulation models and in the development of surrogate models are crucial for accurate extraction of the EOS from intermediate-energy heavy-ion collision data.

The present work therefore primarily addresses the issue with a deep learning (DL)-based solution for accelerating the event-by-event simulation of heavy-ion collisions. Deep learning techniques have been widely used for various physics studies of relativistic heavy-ion collisions [41–56]. Recently, generative modeling, an unsupervised deep learning technique that learns to generate realistic synthetic samples based on patterns learned from the training data, has gained considerable attention [57,58]. Generative models have served successfully in several tasks, such as natural language processing and image synthesis.

Currently, generative methods are also being explored in heavy-ion physics to generate various final-state spectra [59,60] and the calorimeter response in experiment [61]. While these methods are useful in specific scenarios, our goal is to develop a model that generates a complete, event-by-event, heavy-ion collision output. Such a framework can be used for rapid computation of any observable as predicted by a given theoretical model, thus eliminating the need to train separate surrogate models for each observable. This advantage is particularly valuable in comprehensive Bayesian analyses, where numerous, computationally intensive observables (e.g., higher-order fluctuations or multidifferential flow) must often be considered for constraining the EOS. Moreover, such a generative framework can enable real-time model-data comparisons during experiments, and can be combined with optimization techniques such as gradient descent to identify model inputs that best describe the data, offering fast and robust alternatives for exploiting large amounts of experimental data and extracting various properties of the system.

This work lays the groundwork for an ultrafast artificial intelligence (AI)-driven event-by-event simulation model for relativistic heavy-ion collisions. As a first step, we illustrate how this goal can be achieved using training data simulated from the UrQMD cascade model for a fixed beam energy and centrality. Complementing the main results of our method presented in Ref. [62], a detailed account of the model structure, its implementation, and a comprehensive analysis of its performance are presented here.

II. THE URQMD MODEL AND POINT-CLOUD REPRESENTATION

UrQMD provides a well-established nonequilibrium transport description of relativistic heavy-ion collisions over a wide range of collision energies, from SIS to RHIC and LHC. Within UrQMD, hadrons are propagated according to Hamilton's equations of motion and interact via stochastic binary scatterings, color string formation, and resonance excitation and decays. The default setup of UrQMD is a hadronic cascade model which has the effective EOS of a hadron resonance gas (HRG) [63]. There is an option to turn on nuclear potentials and an intermediate hydrodynamic stage to include arbitrary EOS in the evolution of the system. However, for the present demonstrative study, we restrict ourselves to the cascade mode of UrQMD.

In UrQMD, both projectile and target nuclei are boosted toward each other with impact parameter b less than the sum of the projectile and target radii, i.e., $b < R_p + R_t$. The initial positions of nucleons in each nucleus follows a Woods-Saxon radial distribution. The nucleons are assigned random Fermi momenta. A geometric interpretation of the cross section is employed to describe the interactions of hadrons traveling in straight trajectories. The UrQMD cascade describes successfully the final-state hadron spectra over a wide range of energies and the projectile and target masses.

The UrQMD cascade propagates all the nucleons and produces hadrons until all elastic and inelastic interactions cease. Then it outputs the complete list of all final-state particles including their momentum information. An event output from UrQMD can therefore be considered a point cloud in phase space of final-state particles. A point cloud is a collection of points in some feature space with no specific order. Hence, an AI model that aims to generate "UrQMD-like" events must deliver precise and complete particle-by-particle information in such a point-cloud format rather than aggregate information in the form of histograms or spectra.

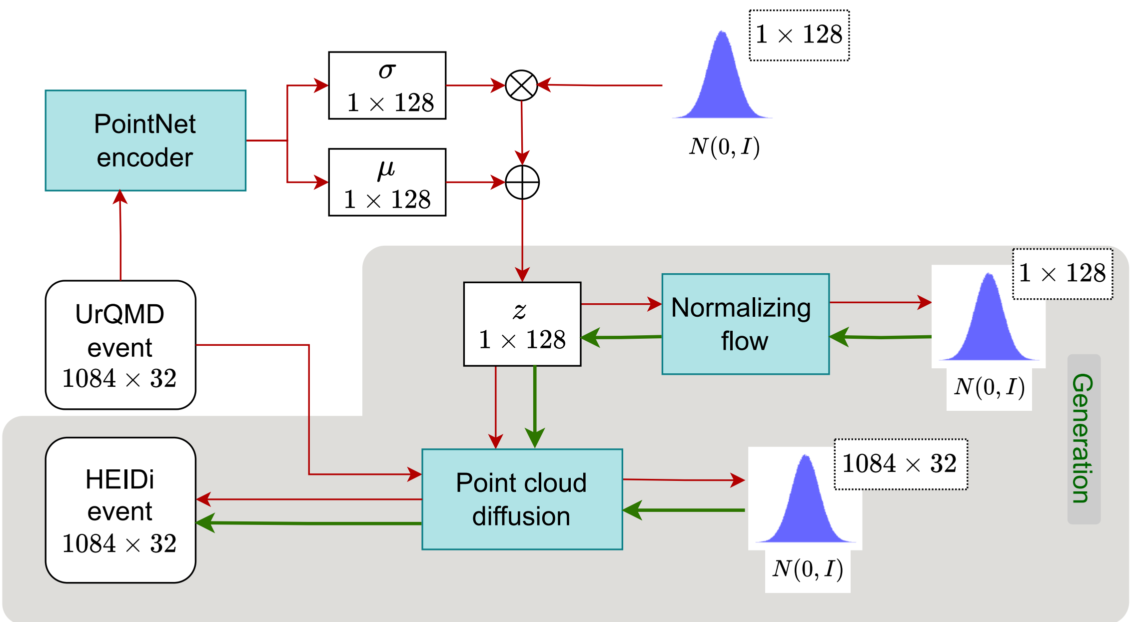

FIG. 1. HEIDi: network structure. The red and green arrows show the flow of information during training and generation, respectively. Only the parts shaded in gray are used during the generation process. A collision event output is represented by a point cloud containing 1084 points, where each point is a final-state particle. σ and μ represent the mean and standard deviation, respectively, of a 128-dimensional Gaussian distribution used to construct the latent condition vector z. The generation process starts with random samples from the standard normal distribution $\mathcal{N}(0, I)$ (mean, zero vector; covariance, identity matrix I), which is processed by the normalizing flow-based decoder and the diffusion model to generate the final-state event output point cloud. The labels "1×128" and "1084×32" refer to the dimensionality of the standard normal distribution from which the random samples are drawn.

III. HEIDI: HEAVY-ION EVENTS THROUGH INTELLIGENT DIFFUSION

Generative modeling of point clouds is an active area of research in AI, motivated by their extensive use in fields where data naturally take the form of unordered point sets. Electronically collected data often have an inherent point-cloud structure. Point clouds are the natural form for representing LIDAR data, which are used widely in three-dimensional (3D) terrain mapping and autonomous driving, for 3D medical scans in medical imaging, and for processing any sensor-based data in general. Generative models for point clouds find applications in point-cloud completion, synthesis, denoising, and segmentation, altering drastically how electronic data are processed across various research fields. From generative adversarial networks to autoregressive point-cloud generation, point-cloud diffusion, and flow-based models, different approaches for generating point clouds have been proposed [64–68].

Diffusion models, in particular, start by sampling from a simple well-known distribution and then iteratively denoise them to generate synthetic samples from the original data distribution. Diffusion models have been explored in particle physics for generating calorimeter responses [69–73], jets [74–79], and electron-proton scattering events [80,81] at high-energy particle collider facilities worldwide.

In this work, we present a probabilistic, conditional diffusion model based on Ref. [67] to generate point clouds representing the complete heavy-ion collision event output, hereafter referred to as HEIDi (Heavy-ion Events through Intelligent Diffusion).

HEIDi is trained here by using 18 000 Au-Au collision events at $E_{\text{lab}} = 10A$ GeV with fixed impact parameter $b = 1$ fm. In order to fix the input dimensions of the training data, each event is represented as a point cloud $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{1084}$ containing 1084 particles (points), where 1084 is a number larger than the maximum event multiplicity of the training dataset. Events with multiplicity less than 1084 are filled up with zeros, which denote fake particles that maintain consistent input dimensions. Each particle $\mathbf{x}_i$ in the point cloud has 32 attributes, of which 3 attributes represent the three components of the momentum vector, p_{x_i}, p_{y_i}, and p_{z_i}, while the remaining 29 attributes denote the one-hot encoded particle information (ID_i):

$$\mathbf{x}_i = \{\mathbf{p}_i, \text{ID}_i\}, \quad \mathbf{p}_i = (p_{x_i}, p_{y_i}, p_{z_i}). \tag{1}$$

The one-hot encoding ID_i creates a 29-dimensional vector denoting unique IDs for the 26 different hadron types $\bar{p}$, $\bar{n}$, K^-, $\bar{K}^0$, $\bar{\Lambda}^0$, $\bar{\Sigma}^-$, $\bar{\Sigma}^0$, $\bar{\Sigma}^+$, $\bar{\Xi}^0$, $\bar{\Xi}^+$, $\bar{\Omega}$, n, p, π^-, π^0, π^+, η^0, K^0, K^+, Λ^0, Σ^-, Σ^0, Σ^+, Ξ^-, Ξ^0, and Ω^-, as well as IDs for spectator[1] neutrons n', spectator protons p', and the fake particles 0. In this way we create a point cloud of complete collision event output represented as a 1084×32 array for each event.

The structure of HEIDi is illustrated in Fig. 1. HEIDi comprises two major components: (1) a PointNet-based [82]

[1]Nucleons which did not undergo any elastic or inelastic interaction are defined as spectator nucleons.

encoder, with a normalizing flow-based [83] decoder, for generating the condition vector z that encodes various global event features and correlations, and (2) a diffusion model [84] that takes the vector z as a condition to generate a point cloud of collision output.

Our first objective is to generate a condition z that informs the diffusion model of various correlations or features that should be present in the generated data. For this, we need a way to map various correlations in an event in the training data into a latent variable z and also learn the probability distribution of this latent variable. After training, by sampling from this probability distribution, we can then generate the condition z. This is achieved by the use of a PointNet-based encoder and a normalizing flow-based decoder.

A. PointNet-based encoder

PointNet is a network designed to learn efficiently from point-cloud data while respecting its permutation invariance. Applications of PointNet-based models in heavy-ion collisions have already been explored in several works [42–44,85,86]. PointNet constructs order invariant features of a point cloud by extracting features of each point separately and converting them into a global feature of the entire point cloud, by using a symmetric mathematical function such as a sum or average of all the pointwise features:

$$\mathcal{Y} = f(\mathbf{X}) = g(h(\mathbf{x_1}), h(\mathbf{x_2}), \ldots, h(\mathbf{x_{1084}})). \tag{2}$$

Here, $\mathcal{Y}$ is an order-invariant feature of the point cloud, h is the pointwise feature extraction operation usually modeled as a neural network, using one-dimensional (1D) convolution operations or a multilayered perceptron with shared weights. g is the symmetric operation applied on the extracted pointwise features.

For every UrQMD event in the training dataset, our PointNet-based encoder predicts two global feature vectors of dimensions 1×128 each, which represent the mean μ and the standard deviation σ of a 128-dimensional Gaussian distribution. Random samples generated from this Gaussian $\mathcal{N}(\mu, \sigma)$ have to be used as the condition vector for the diffusion model to optimize the encoder parameters and learn the correct encoding. To achieve this, the gradients of loss or objective function $\mathcal{L}$ (a mathematical function which quantifies the difference between the expected and actual outputs) with respect to model parameters needs to be computed. Since random sampling is nondeterministic and nondifferentiable, the well-known reparametrization trick [87] is used to ensure the flow of gradients through the network. Here, the random sampling process is isolated into a fixed distribution, typically a standard normal distribution $\mathcal{N}(0, I)$ (mean, zero vector; covariance, identity matrix I) and the latent encoding is parametrized as the variable z, given by

$$z = \mu + \sigma \cdot \epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

The encoder neural network with parameters ϕ estimates the approximate probability distribution $\tilde{q}_\phi(z|\mathbf{X})$ for condition z given the point cloud $\mathbf{X}$ of UrQMD collision output as a Gaussian with mean μ_ϕ and covariance Σ_ϕ predicted by the network, i.e.,

$$\tilde{q}_\phi(z|\mathbf{X}) = \mathcal{N}(z|\mu_\phi(\mathbf{X}), \Sigma_\phi(\mathbf{X})). \tag{4}$$

B. Normalizing flow-based decoder

To complete the pipeline, which involves generating the conditional vector z, we still need to know the probability density $p(z)$. We use a normalizing flow-based model to facilitate the sampling of z from $p(z)$. Instead of directly learning the potentially complex, high-dimensional $p(z)$ distribution, the normalizing flow-based model with parameters α learns bijective transformations $\mathcal{B}_\alpha(z)$ which convert samples from $p(z)$, as generated by the PointNet-based encoder, into samples w following a simple, well-known distribution, such as the standard normal distribution $\mathcal{N}(0, I)$. Thus, after training, we can invert the process to sample from the prior normal distribution and transform it via $\mathcal{B}_\alpha^{-1}(w)$ into samples from $p(z)$. For more details on normalizing flow models, see Ref. [88]. The bijective nature of the transformations modeled by this neural network would allow us to invert the transformations and extract the exact probability for z as

$$p(z) = p(w) \cdot \left| \det \frac{\partial \mathcal{B}_\alpha^{-1}}{\partial w} \right|^{-1}, \tag{5}$$

where $w = \mathcal{B}_\alpha(z)$ and $\det \frac{\partial \mathcal{B}_\alpha^{-1}}{\partial w}$ is the Jacobian determinant of $\mathcal{B}_\alpha^{-1}$.

C. Point-cloud diffusion

The final step is to generate a point cloud of collision output based on the condition z. For this, we employ a probabilistic diffusion model. For different implementations and applications of diffusion models, see Refs. [89–92]. The learning process in a diffusion model comprises a forward diffusion step and a reverse antidiffusion step.

In the forward diffusion step, a controlled Gaussian noise is gradually added to the UrQMD data over several time steps until the point cloud is transformed into pure noise. The original UrQMD-generated point cloud $\mathbf{X} = \mathbf{X}^0 = \{\mathbf{x}_i^0\}_{i=1}^{1084}$ is considered as the initial, undisturbed state of the point cloud at time step $t = 0$. At each subsequent time step, a Gaussian noise is added to the previous state of each point in the point cloud to obtain the current state of the point. This results in states $\mathbf{x}_i^0, \mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^T$, where T is the final time step. By gradually increasing the noise level in the points through the forward process, the final state of the point cloud $\mathbf{X}^T = \{\mathbf{x}_i^T\}_{i=1}^{1084}$ eventually resembles samples from a standard normal distribution.

Given the state of a point $\mathbf{x}_i^{t-1}$ at time step $t-1$, the probability distribution $q(\mathbf{x}_i^t|\mathbf{x}_i^{t-1})$ for the next state $\mathbf{x}_i^t$, after addition of noise at time step t, is described by a Gaussian kernel

$$q\big(\mathbf{x}_i^t\big|\mathbf{x}_i^{t-1}\big) = \mathcal{N}\big(\mathbf{x}_i^t\big|\sqrt{1-\beta_t}\mathbf{x}_i^{t-1}, \beta_t\mathbf{I}\big) \tag{6}$$

with a mean of $\sqrt{1-\beta_t}\mathbf{x}_i^{t-1}$ and variance of $\beta_t\mathbf{I}$. Here β_t is a hyperparameter controlling the amount of noise added at each time step and $\mathbf{I}$ is the identity matrix with dimensions matching the particle vector $\mathbf{x}_i$.

The reverse antidiffusion process is the inverse of the forward process which transforms Gaussian noise samples to the final point cloud. Unlike the forward diffusion process, the reverse process requires training a neural network to identify and remove the noise added to the state in each corresponding forward step. Starting with a sample from the standard normal noise distribution $\tilde{q}(\mathbf{x}_i^T) = \mathcal{N}(0, I)$, which approximates $q(\mathbf{x}_i^T)$, a neural network sequentially predicts the denoised state in the previous time step, finally recovering the original point $\mathbf{x}_i^0$ in the point cloud.

Given the state $\mathbf{x}_i^t$ and the condition vector z, the probability for the previous state $\mathbf{x}_i^{t-1}$ is also given by a Gaussian kernel,

$$\tilde{q}_\theta\left(\mathbf{x}_i^{t-1}\middle|\mathbf{x}_i^t, z\right) = \mathcal{N}\left(\mathbf{x}_i^{t-1}\middle|\mu_\theta\left(\mathbf{x}_i^t, t, z\right), \beta_t I\right), \tag{7}$$

where μ_θ is the denoised mean as predicted by a neural network with parameters θ, based on the current state $\mathbf{x}_i^t$, the time step t, and the latent vector z.

D. Training and generation

The PointNet-based encoder, the normalizing flow-based decoder, and the probabilistic diffusion model are trained together to minimize the objective function given by

$$\mathcal{L} = \mathbb{E}_q\left[\sum_{t=2}^{T}\sum_{i=1}^{N} D_{\mathrm{KL}}\left(q\left(\mathbf{x}_i^{t-1}\middle|\mathbf{x}_i^t, \mathbf{x}_i^0\right)\middle\|\tilde{q}_\theta\left(\mathbf{x}_i^{t-1}\middle|\mathbf{x}_i^t, z\right)\right) - \sum_{i=1}^{N}\ln \tilde{q}_\theta\left(\mathbf{x}_i^0\middle|\mathbf{x}_i^1, z\right) + D_{\mathrm{KL}}(\tilde{q}_\phi(z|\mathbf{X}^0)\|p(z))\right]. \tag{8}$$

Here, $\mathbb{E}_q$ stands for the expectation value with respect to the distribution of q and D_{KL} stands for the Kullback-Leibler divergence, which quantifies the difference between the two distributions.

The first term measures the KL divergence between the true posterior for state $q(\mathbf{x}_i^{t-1})$, given previous state $\mathbf{x}_i^t$, and the undisturbed UrQMD data $\mathbf{x}_i^0$ and the posterior approximated by the diffusion model $\tilde{q}_\theta(\mathbf{x}_i^{t-1})$, given the state $\mathbf{x}_i^t$ and condition vector z. As discussed in Ref. [84], the true posterior for a point can be computed to be a Gaussian whose mean and variance depend only on the undisturbed state $\mathbf{x}_i^0$, current state $\mathbf{x}_i^t$, and the hyperparameter β_t that controls the noise added in each step.

Equation (7) gives the approximate posterior $\tilde{q}_\theta(\mathbf{x}_i^{t-1}|\mathbf{x}_i^t, z)$. The summation iterates over all time steps, from $t = 2$ to T, and across all points, $i = 1$ to N, computing the KL divergence for the entire point cloud across all time steps.

The second term calculates the negative log-likelihood of the original UrQMD point cloud, given the noisy point cloud at time step $t = 1$ and the condition vector z, which is given also by Eq. (7).

The last term in the objective function calculates the KL divergence between the posterior for condition z, given the UrQMD event $\mathbf{X}^0$, and is given by Eq. (4). Finally, the probability distribution of z is given by Eq. (5).

For more details on the loss function and the training algorithm, we refer to the original implementations of the probabilistic diffusion model [84] and the point-cloud diffusion model [67].

After successful training, only the normalizing flow-based decoder and the diffusion models are necessary for the generation of point clouds. This part of the network is highlighted with gray background in Fig. 1. The PointNet encoder is necessary only when training the model.

IV. RESULTS

HEIDi is trained to minimize the objective function defined in Eq. (8). After training, the model is evaluated for its capability to generate "realistic collision events" as described by UrQMD. To assess this, various distributions of distinct particle species from 50 000 HEIDi-generated events are compared with 50 000 UrQMD-generated events.

HEIDi generates a list of 1084 particle vectors per event, where each vector comprises the three-dimensional momentum vector and the unique ID that identifies the respective hadron species. Removing all the fake particles (ID = 0) in a generated event gives the final-state output particles in that event. Although HEIDi generates particles from all 26 different hadron types, for these comparisons we focus on the 16 most abundant hadron species. The remaining 10 hadron types have too-low event multiplicities (i.e., mean multiplicities $\lesssim 10^{-2}$/event).

A comparison of the p_x, p_y, and p_z distributions for various particle types generated by HEIDi and UrQMD are shown in Figs. 2 and 3. Evidently, the momentum distributions of various hadron types for particles generated by HEIDi are in excellent agreement with those by UrQMD simulations. Despite slightly underestimating very small values of p_x and p_y, of spectator nucleons, HEIDi has successfully learned the p_z distributions of spectators, which differ drastically from the distributions of participant nucleons and from all other hadron species.

Apparently, HEIDi performs well in learning the individual components of the momenta of various hadron species. It is interesting to see how well the joint probabilities of the momentum components are learned. Figure 4 shows the the transverse momentum distributions of various hadrons. HEIDi learns accurately all p_T distributions for different hadrons. With only a moderate training dataset, HEIDi efficiently captures the complex joint probability distributions, across *six orders of magnitude* for all hadron species.

In addition to learning various features of individual particles generated in each event, it is very important that HEIDi captures various global properties of events such as the specific mean transverse momenta $\langle p_T \rangle$ and the average multiplicities of each hadron species separately in each and every event. The $\langle p_T \rangle$ distributions of various hadrons are presented in Fig. 5.

A good agreement between the $\langle p_T \rangle$ distributions of HEIDi events and UrQMD events for all different hadron species can be noted. For certain particle species and at very high $\langle p_T \rangle$, where the training data samples are rather limited, HEIDi learns the distribution so well that only moderate deviations from UrQMD distributions are observed.

Figure 6 presents the multiplicity distributions of different hadron species. HEIDi distinguishes the differences in the multiplicities of various hadron types in each event, accurately learning the relative probabilities. HEIDi also recognizes the

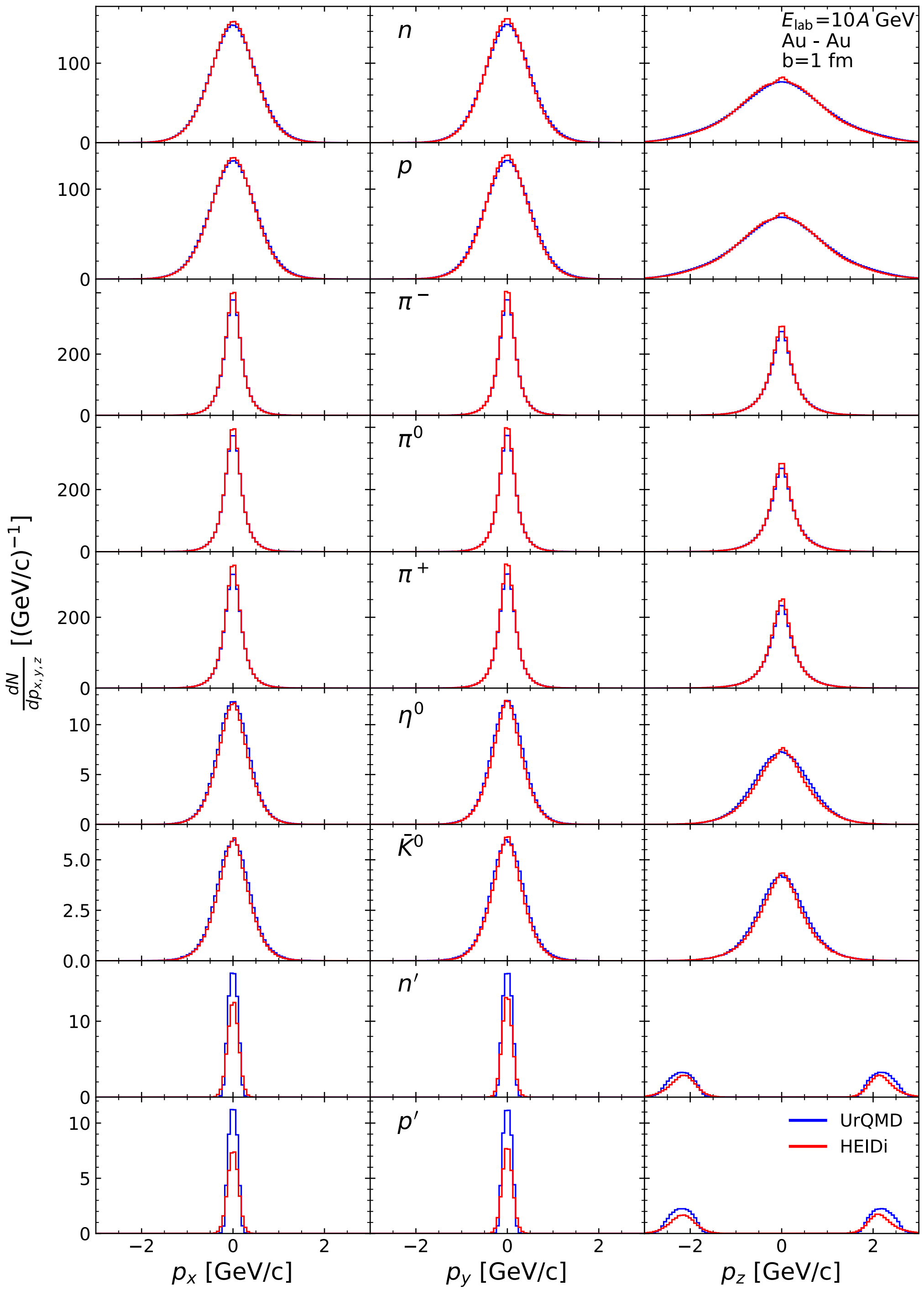

FIG. 2. Momentum distributions of various hadron types generated by HEIDi for Au-Au collisions with $b = 1$ fm at 10A GeV. Each row corresponds to one hadron type. Columns 1, 2, and 3 represent the distributions for p_x, p_y, and p_z, respectively. UrQMD and HEIDi results are shown in blue and red curves, respectively.

large difference in the multiplicity distributions for spectators and participants, thus well modeling the event-by-event fluctuations of the respective interaction volumes. For certain hadron types, HEIDi slightly overestimates the probability for very high and very low multiplicities.

Reproducing the multiplicity distributions are much more difficult than reproducing the rapidity distributions or momentum distributions, as it requires accurately capturing small, many-particle correlations in individual events. The event-by-event multiplicity of the particles which show these deviations such as neutrons, protons, and pions also have significantly higher variance compared to those of rarer particles. Learning high-variance distributions of more abundant particles from a limited dataset is more challenging as here the many-body correlations are more important statistically. If a particle is very rare, then its multiplicity distribution will approach the

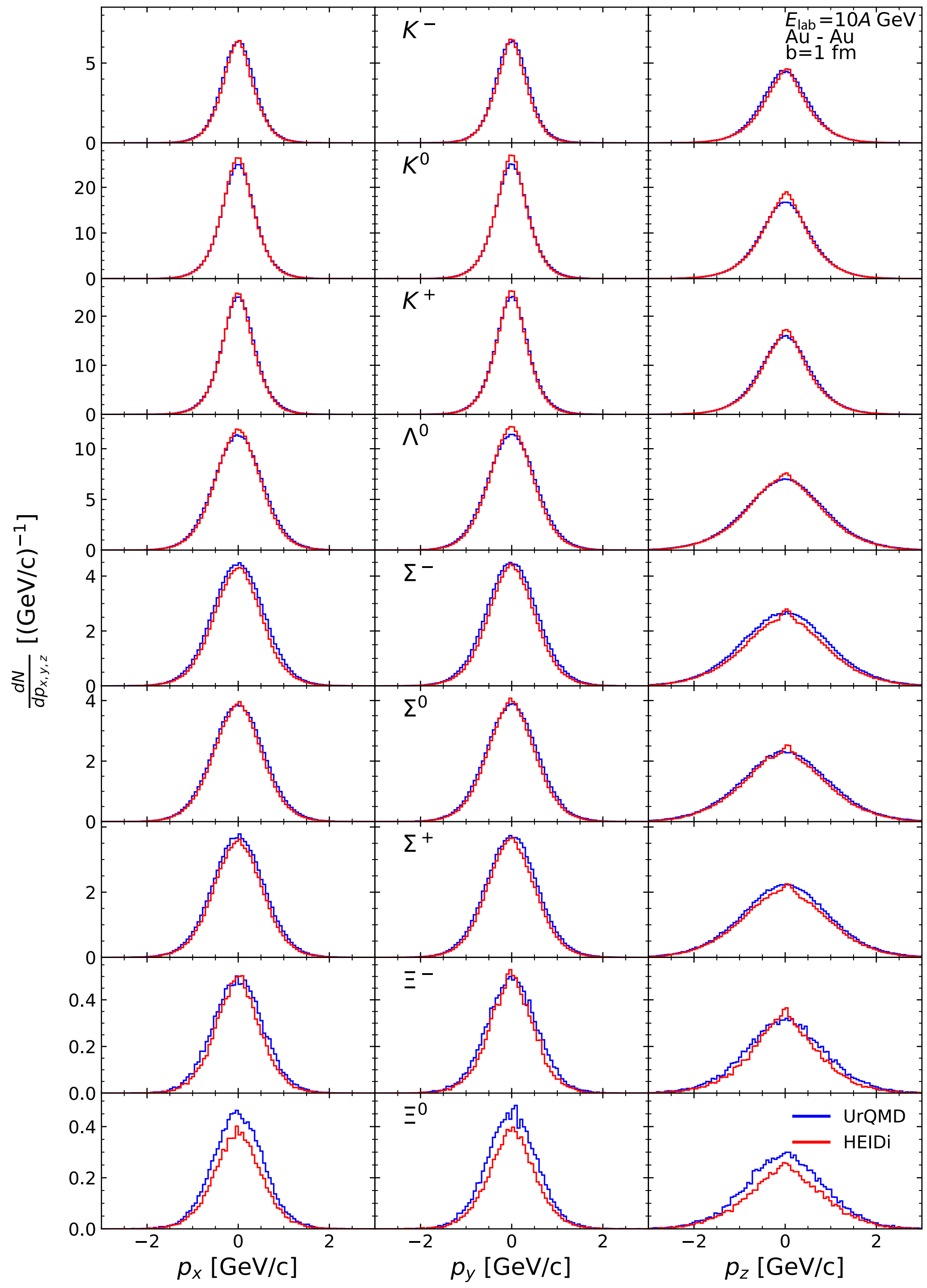

FIG. 3. Momentum distributions of various hadron types generated by HEIDi for $b = 1$ fm, Au-Au collisions at 10A GeV. The color scheme and plot layouts are similar to Fig. 2.

Poissonian limit, which is easy for a model to learn. For particles that are very abundant, the contribution from many-particle correlations becomes more and more prevalent and thus is harder to capture. Although our model is designed to capture such correlations, the diffusion model is informed of these correlations by the condition vector generated by the normalizing flow decoder rather than hard constraints. Thus, it is easier for the model to learn the less-correlated low-variance distributions of the rare particles. We also find that increasing the diversity of training data by conditioning the model on different collision centralities significantly improves the overall multiplicity distributions of these abundant particles for various centralities. This is discussed in Sec. V.

It is important to emphasize that generative models are usually not benchmarked on the multiplicity distributions but rather on being able to create either single instances or event-

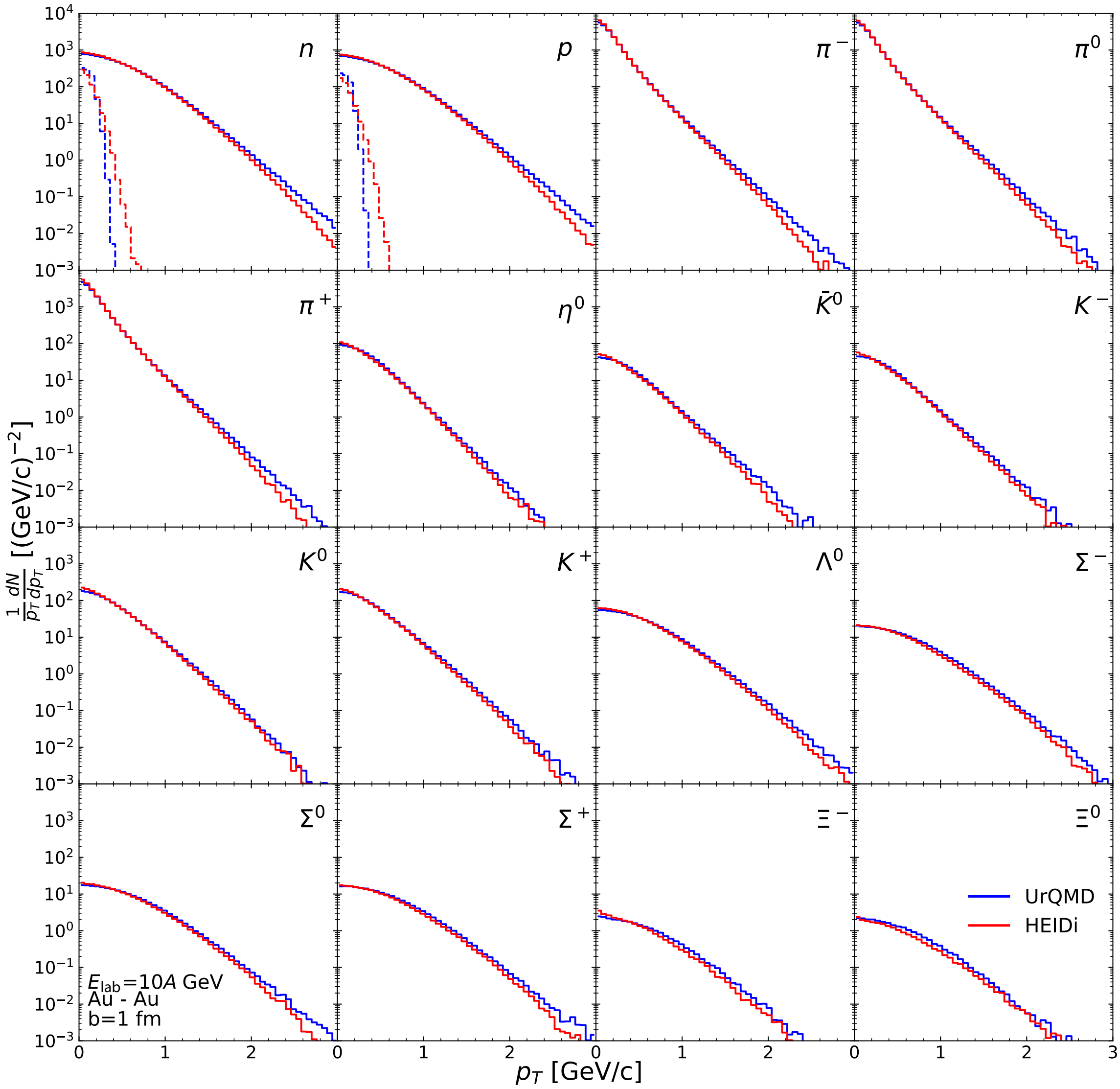

FIG. 4. Transverse momentum distribution of various hadrons. The results are for 10A GeV Au-Au collisions with impact parameter $b = 1$ fm. The HEIDi distributions are shown as red curves while the UrQMD distributions are shown in blue. The dashed curves in the upper left two plots are the p_T distributions for the spectator nucleons, where the solid curves in the plots show the p_T distributions for participant nucleons.

averaged distributions. In our work, this has already been demonstrated in Fig. 5, where we observe excellent agreement with UrQMD over more than five orders of magnitude in probability for all considered particle species.

The rapidity distributions of various hadrons generated by HEIDi are compared with those from UrQMD simulations in Fig. 7. The particles generated by HEIDi follow the rapidity distributions of hadrons in UrQMD data well. For some hadrons, a slight but visible excess of particles is found in HEIDi events for small rapidities and p_z close to zero. HEIDi tends to overestimate the number of very low momentum particles as their number in the current training dataset is quite limited. However, this only leads to a small effect on the total yield.

We have already demonstrated that the model accurately reproduces the marginal distributions of various features, as well as various momentum space correlations, as reflected in the rapidity and transverse momentum distributions. To further assess the model's ability to capture event-level correlations, we investigate the event-by-event multiplicity correlation between protons and π^+, as shown in Fig. 8. The model successfully captures the overall negative correlation between the proton and π^+ multiplicities in an event. While the learned correlation does not perfectly match the true correlation from UrQMD, this is expected, as accurately learning detailed event-by-event correlations remains a challenge for generative models. These deviations in learned correlations, in turn, contribute to the discrepancies observed in the multiplicity distributions presented in Fig. 6.

Despite being trained on a relatively small dataset of 18 000 events, HEIDi successfully captured most of the key properties and correlations among distinct hadron types, as well as global event characteristics. With this limited dataset, HEIDi is able to generate "UrQMD-like events" with the components

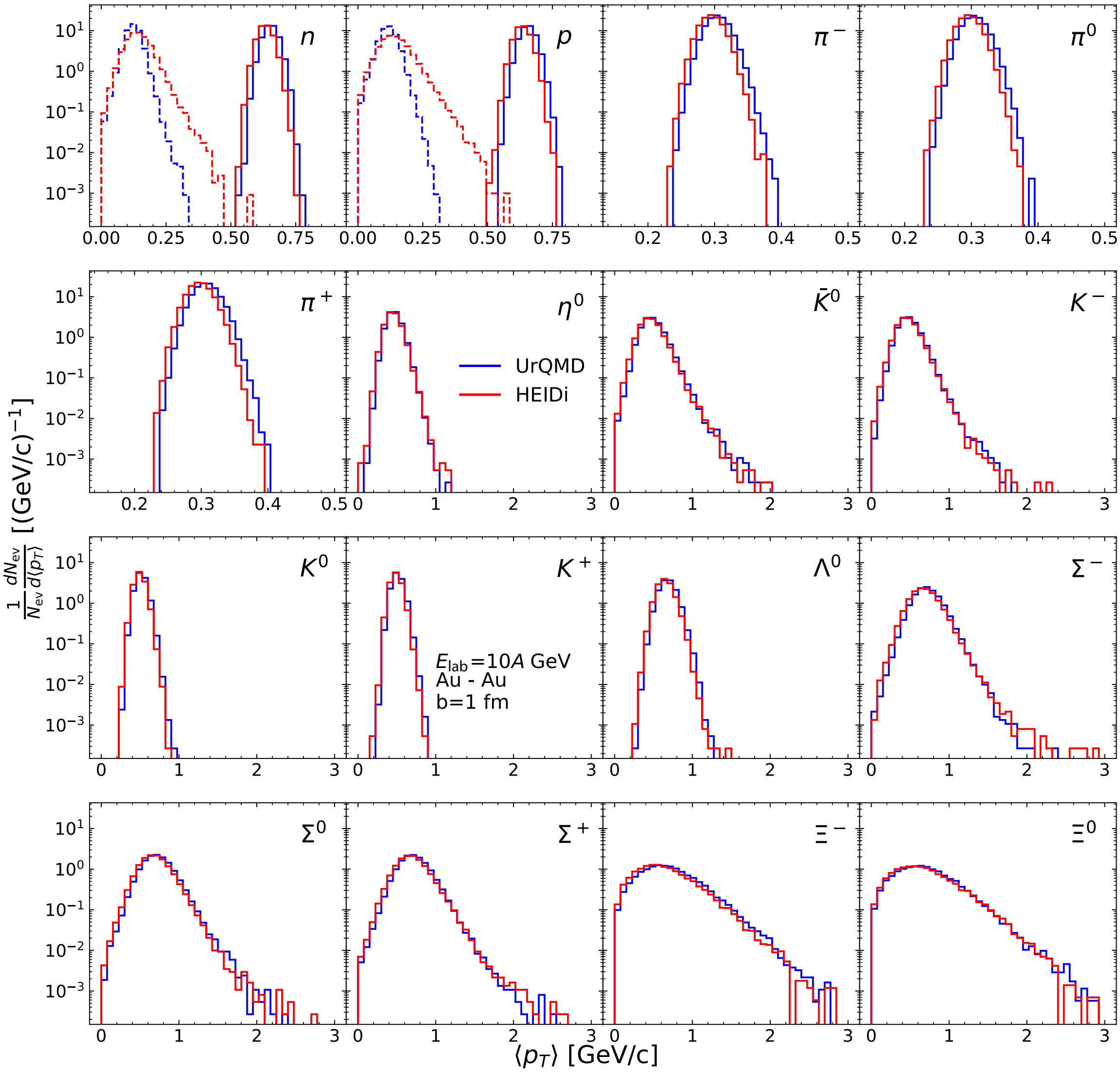

FIG. 5. $\langle p_T \rangle$ distribution of various hadrons for Au-Au collisions with $b = 1$ fm at $10A$ GeV. The HEIDi distributions are shown as red curves, while the UrQMD distributions are shown in blue. The dashed curves in the upper left two plots are the distributions for spectator nucleons and the solid curves in the plots represent the distributions for participant nucleons.

of momentum vector or the transverse momentum of generated particles closely following the UrQMD distributions even in the tails where the training samples are sparse. The small dataset was chosen to demonstrate HEIDi's ability as an AI emulator to learn underlying probability distributions and correlations from limited samples to ultimately generate a large number of realistic samples quickly to achieve a higher statistical significance. However, capturing all correlations in UrQMD, particularly very rare or absent ones in the training data, is inherently difficult. This limitation is reflected in discrepancies in the tails of the multiplicity distribution and in the number of very low-momentum particles, indicating regions of phase space where the model might be undertrained. A better training strategy or a diverse training set would be necessary to improve the accuracy of the model in these regions.

The primary motivation to develop a neural-network-based event-by-event emulator for the known traditional heavy-ion collision simulation models is to overcome the computational bottleneck due to the large computation times which prevent us from performing necessary, expensive parameter estimation tasks. When deployed on an Nvidia A100 GPU with 40 GB of memory, HEIDi generates an event in about 30 ms, whereas the UrQMD cascade model takes approximately 3 s per event. This speedup of about two orders of magnitude is truly remarkable as it is straightforward to train HEIDi on similar input data from more complex and expensive $(3 + 1)$-dimensional dynamical models, such as UrQMD with potentials, complex in-medium momentum-dependent cross sections, etc., or the hybrid model (including hydrodynamic phase) of UrQMD. Here speedups of at least five orders of magnitude are expected. HEIDi, trained on such datasets, will be fast enough for a comprehensive Bayesian inference of the EOS or any other physics parameters, for which numerous, multidifferential observables will necessarily be calculated. HEIDi is fully differentiable: the gradients of the objective function with respect to every single model parameter can

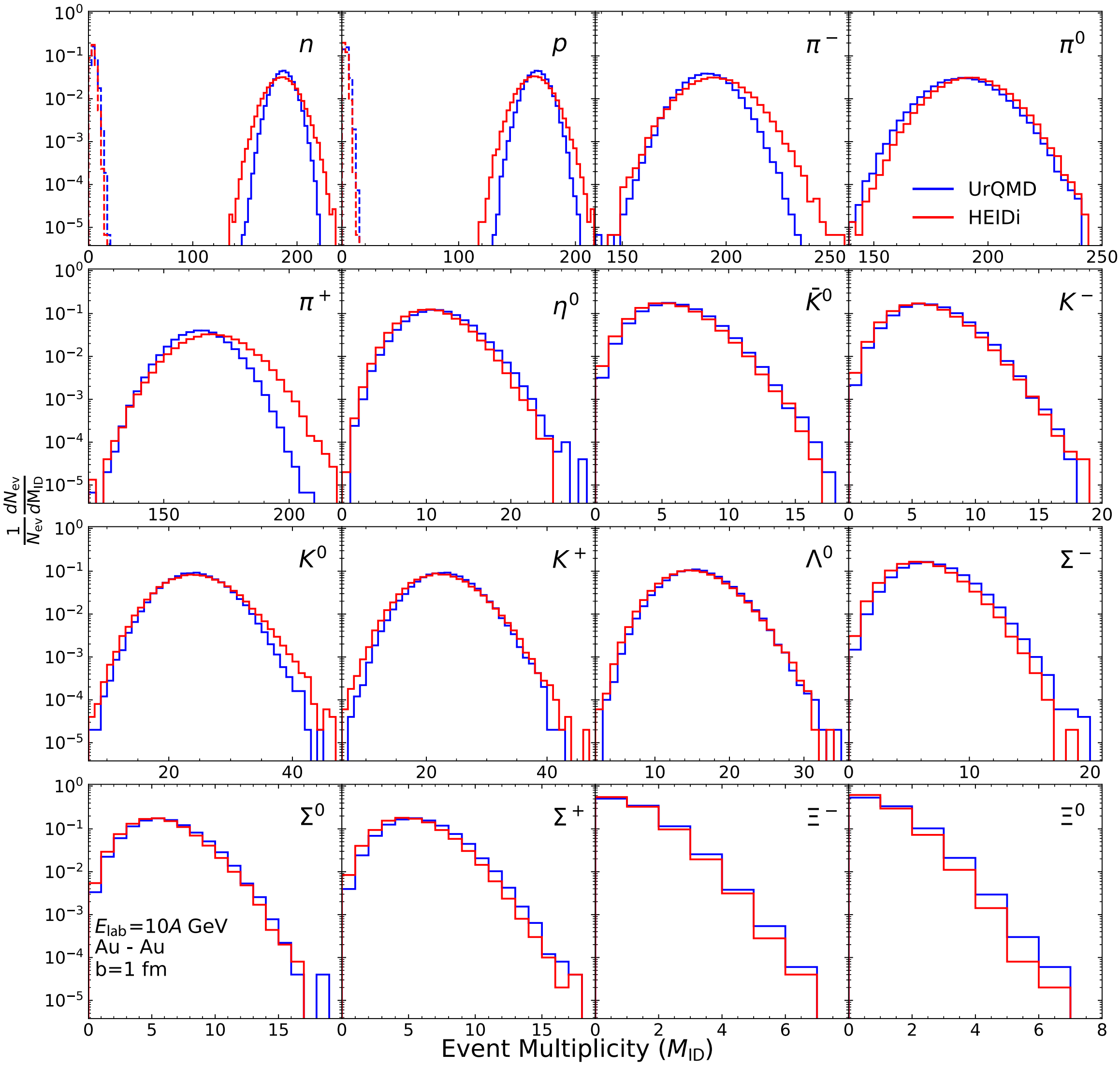

FIG. 6. Multiplicity distribution of various hadron species for Au-Au collisions with $b = 1$ fm at 10A GeV. In the upper left two plots, the distributions for spectator nucleons are shown as dashed curves while the solid curves in the plots denote the distributions for participant nucleons. The multiplicity distribution constructed from HEIDi events are shown as red curves while the distributions from UrQMD events are shown in blue.

be calculated. Hence, novel optimization techniques can be explored to extract the underlying model parameters directly from event-by-event experimental data, without the use of predefined observables.

V. CONDITIONING THE MODEL ON COLLISION PROPERTIES

For the model to be practically useful in tasks such as Bayesian inference, the model should be able to generate collision events with varying properties such as beam energy, centrality, and the equation of state. We have already demonstrated that the model accurately generates collision event outputs for fixed-centrality ($b = 1$ fm) Au-Au collisions at 10A GeV. Extending the model to handle events with different collision features is straightforward and can be achieved by extending the latent vector z with necessary collision properties. The required condition (collision property) can be added as an additional nontrainable parameter to the latent vector z generated by the normalizing flow decoder.

To illustrate this capability, we trained the HEIDi architecture on 30 000 Au-Au events at 10A GeV, comprising 10 000 events each for impact parameter of 1, 3, and 5 fm. The impact parameter was included as a conditioning variable by extending the latent vector z accordingly. The model was then tested on UrQMD events with impact parameters of 1, 3, 4, and 5 fm. The resulting multiplicity distributions for selected hadron species are shown in Fig. 9.

It can be seen that the model successfully captures the centrality dependence of hadron multiplicities. While some discrepancies remain at $b = 1$ fm, there is a good agreement with UrQMD results for the other tested impact parameters. Furthermore, the model also demonstrates strong interpolation capabilities, as evident in the accurate

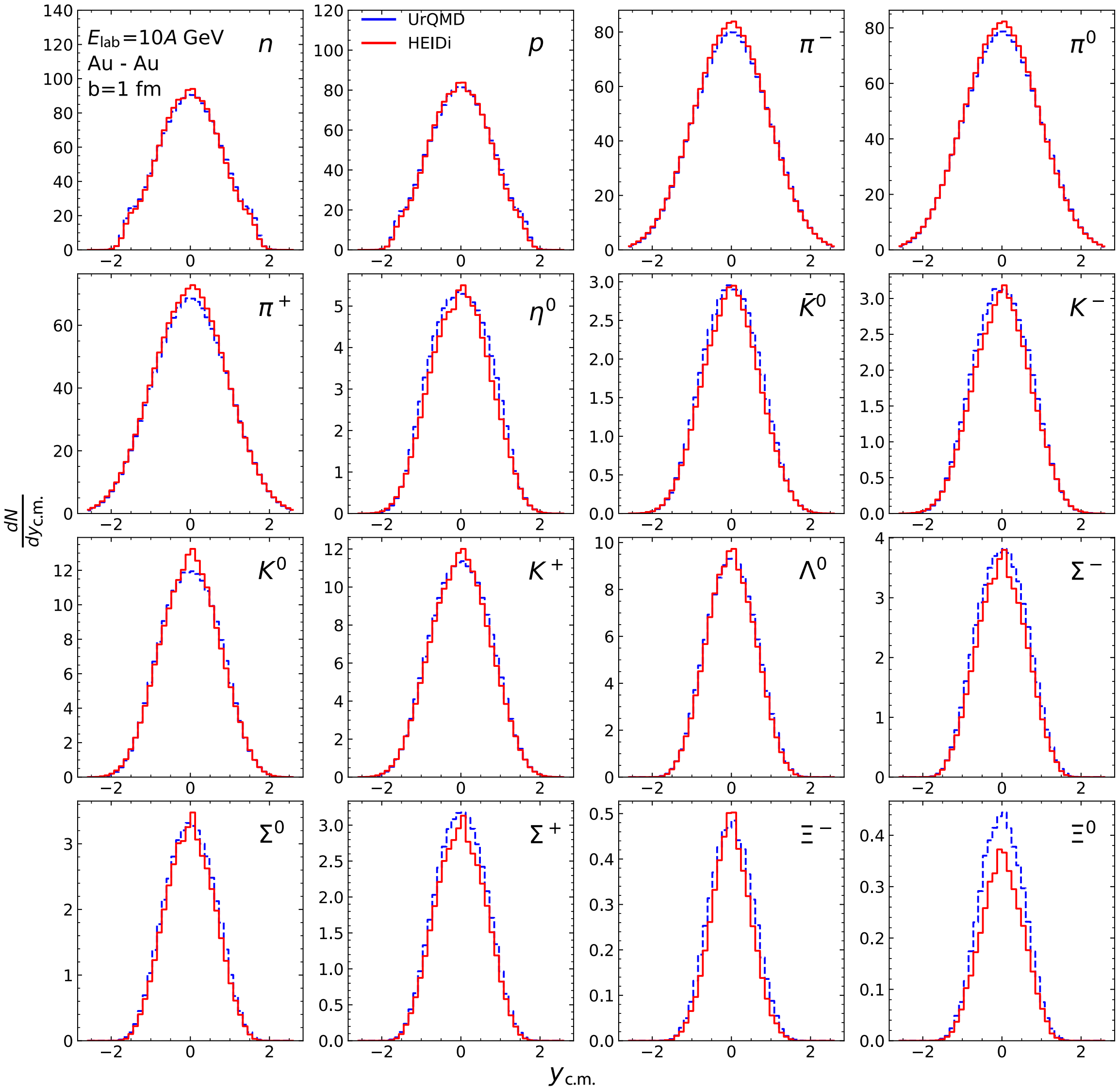

FIG. 7. Rapidity distribution of various hadron species for Au-Au collisions with $b = 1$ fm at 10A GeV. The distributions constructed from HEIDi events are shown as red curves while the distributions from UrQMD events are shown in blue. The upper left two plots show the rapidity distributions for all neutrons and all protons (including spectators), respectively.

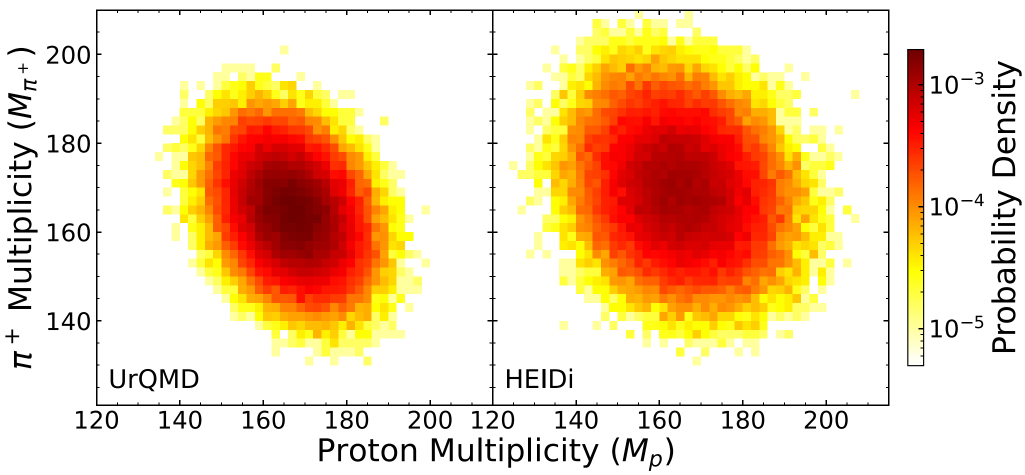

FIG. 8. Event-by-event multiplicity correlation between protons and π^+ in Au-Au collisions for $b = 1$ fm at 10A GeV: (left) the true correlation as obtained from UrQMD events, and (right) the corresponding correlation learned by HEIDi.

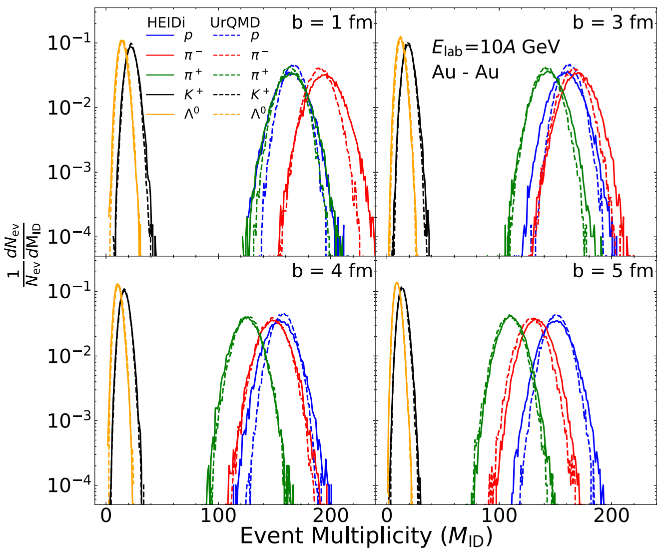

FIG. 9. Multiplicity distributions of selected hadron species for Au-Au collisions at 10A GeV for impact parameters $b = 1$ fm (top left), 3 fm (top right), 4 fm (bottom left), and 5 fm (bottom right). Distributions generated from HEIDi events are shown as solid curves while the distributions generated from UrQMD events are shown as dashed curves.

reproduction of multiplicities at $b = 4$ fm, a value not included in the training set. The corresponding rapidity distributions for selected hadron species across various impact parameters are presented in Ref. [62] and show an excellent agreement between HEIDi and UrQMD for all tested centralities. These results indicate that training the model on a more diverse and extensive dataset enhances the model's generalization ability and predictive power. Furthermore, they demonstrate the flexibility and adaptability of the proposed model in handling additional conditions such as various collision parameters without any modification to the core architecture.

VI. APPLICATIONS IN HEAVY-ION PHYSICS AND BEYOND

HEIDi as a robust framework for comprehensive Bayesian inference or other parameter estimation tasks can be deployed for various important applications within high-energy heavy-ion collisions and other detector-intensive experiments. HEIDi uses a point-cloud data structure, which is extremely flexible and easily adaptable for diverse tasks in high-energy physics. Developers of different models or detector simulations can readily adapt HEIDi to build their own generative AI simulation frameworks trained on specific parameter sets or physical features, thereby enabling rapid generation of large-scale event samples without requiring detailed knowledge of the underlying code of the physics model. In addition, experimental collaborations can deploy models based on HEIDi for fast, online data comparison, which can be useful for first-level event analysis, triggers for interesting physics, calibration, quality control, and real-time monitoring, fault, or anomaly detection in detector systems.

Conditional point-cloud diffusion has a wide range of potential applications beyond heavy-ion physics. Architecture similar to HEIDi can easily perform analyses and generative tasks for scientific data processing wherever data are collected electronically or with multiple sensors with point-cloud structure. HEIDi demonstrates the feasibility of conditional generation for high-dimensional, categorical point clouds. Similar models can find important applications across various other fields which directly benefit mankind, such as biomedical imaging and diagnosis, predictive maintenance for technological hardware, early warning systems for natural hazards, environmental surveillance, urban planning [93], and autonomous systems.

VII. CONCLUSIONS

This work introduces HEIDi, a conditional point-cloud generative model for event-by-event collision output in relativistic heavy-ion collisions. Based on Ref. [67], HEIDi uses a PointNet-based point-cloud encoder and a normalizing flow-based decoder to create a latent encoding which is used as a condition for a probabilistic diffusion model that generates complete collision output. When trained on UrQMD cascade data, the model learns to generate "realistic" point clouds of collision output which contains the 26 most abundant distinct hadron species. Particles generated by the DL model accurately reproduce different probability distributions of UrQMD data. The HEIDi model not only captures various correlations among different particles in each event but also learns well the different event-level properties. The generated hadrons are shown to successfully reproduce the UrQMD distributions, both for different components of momentum vectors p_x, p_y, and p_z, transverse momenta p_T, and multiplicities M as well as rapidities y_{cm}. HEIDi generates events two orders of magnitude faster than the conventional UrQMD cascade computations.

Capturing all correlations within an event in high-energy collisions remains a significant challenge for current point-cloud-based generative models, including HEIDi. This limitation is particularly evident in the deviations observed at the tails of the multiplicity distributions for nucleons and pions. Simply increasing the size of the training dataset does not directly lead to substantial improvements in the multiplicity distributions. However, it is found that training HEIDi on a more diverse and extended dataset by including multiple centrality events improves overall agreement with UrQMD for the event-by-event multiplicity distributions of nucleons and pions for various centralities. The same strategy can be easily extended to incorporate a wide range of physical conditions, including collision energy, system and asymmetry, and parameters governing the EOS, without requiring any changes to the underlying model architecture. In addition, we are currently exploring other techniques to enhance the model's ability to capture correlations from limited training data. These include multistep generation approaches such as first generating particle multiplicities and then sampling the particle momenta and conditioning the model directly on initial-state profiles (e.g., nucleon positions and momenta).

The architecture of HEIDi can be adapted for any other event-by-event collision output generation as possibly dictated by any other physical model. The choice of the point-cloud format to represent the event-by-event collision data makes it flexible for accelerating detector simulations as well. Such extensions of HEIDi will be tackled in future research.

HEIDi's high statistics heavy-ion event generation marks a significant milestone in AI-accelerated event-by-event heavy-ion simulations. Through ultrafast generation of complete heavy-ion events in point-cloud format, HEIDi-based models will allow us to use computationally expensive observables for complex parameter estimation tasks, as well as enable ultrafast large-scale simulations for various theoretical studies and experimental applications of interest. As a robust and flexible generative AI framework, HEIDi lays the ground for a first foundation model for heavy-ion collisions.

ACKNOWLEDGMENTS

This work is supported by the the BMBF under the KISS project (M.O.K., K.Z.), the SAMSON AG (J.S., K.Z.), the CUHK-Shenzhen University development fund under Grants No. UDF01003041 and No. UDF03003041 (K.Z.), and Shenzhen Peacock fund under Grant No. 2023TC0179 (K.Z.) and the Walter Greiner Gesellschaft zur Förderung der physikalischen Grundlagenforschung e.V. through the Judah M. Eisenberg Laureatus Chair at the Goethe Universität Frankfurt am Main (H.S.).

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

[1] H. Stöcker and W. Greiner, Phys. Rep. **137**, 277 (1986).
[2] K. Fukushima and T. Hatsuda, Rep. Prog. Phys. **74**, 014001 (2011).
[3] Y. Aoki, G. Endrodi, Z. Fodor, S. D. Katz, and K. K. Szabo, Nature (London) **443**, 675 (2006).
[4] S. Borsanyi, G. Endrodi, Z. Fodor, A. Jakovac, S. D. Katz, S. Krieg, C. Ratti, and K. K. Szabo, J. High Energy Phys. 11 (2010) 077.
[5] A. Bazavov *et al.* (HotQCD Collaboration), Phys. Rev. D **90**, 094503 (2014).
[6] M. Cheng *et al.*, Phys. Rev. D **77**, 014511 (2008).
[7] A. Bazavov *et al.*, Phys. Rev. D **85**, 054503 (2012).
[8] S. Aoki *et al.* (PACS-CS Collaboration), Phys. Rev. D **79**, 034503 (2009).
[9] G. Agakichiev *et al.* (HADES Collaboration), Eur. Phys. J. A **41**, 243 (2009).
[10] C. Yang *et al.* (STAR Collaboration), Nucl. Phys. A **967**, 800 (2017).
[11] M. M. Aggarwal *et al.* (STAR Collaboration), arXiv:1007.2613.
[12] L. Adamczyk *et al.* (STAR Collaboration), Phys. Rev. C **96**, 044904 (2017).
[13] K. C. Meehan (STAR Collaboration), Nucl. Phys. A **956**, 878 (2016).
[14] T. Ablyazimov *et al.* (CBM Collaboration), Eur. Phys. J. A **53**, 60 (2017).
[15] J. C. Yang *et al.*, Nucl. Instrum. Meth. B **317**, 263 (2013).
[16] V. Golovatyuk, V. Kekelidze, V. Kolesnikov, O. Rogachevsky, and A. Sorin, Eur. Phys. J. A **52**, 212 (2016).
[17] S. A. Bass *et al.*, Prog. Part. Nucl. Phys. **41**, 255 (1998).
[18] M. Bleicher *et al.*, J. Phys. G **25**, 1859 (1999).
[19] W. Cassing and E. L. Bratkovskaya, Nucl. Phys. A **831**, 215 (2009).
[20] Y. Nara, N. Otuka, A. Ohnishi, K. Niita, and S. Chiba, Phys. Rev. C **61**, 024901 (1999).
[21] H. Petersen, J. Steinheimer, G. Burau, M. Bleicher, and H. Stöcker, Phys. Rev. C **78**, 044901 (2008).
[22] M. O. Kuttan, A. Motornenko, J. Steinheimer, H. Stoecker, Y. Nara, and M. Bleicher, Eur. Phys. J. C **82**, 427 (2022).
[23] J. Steinheimer, M. O. Kuttan, A. Motornenko, A. Sorensen, Y. Nara, V. Koch, M. Bleicher, and H. Stoecker, EPJ Web Conf. **276**, 01021 (2023).
[24] J. Steinheimer, T. Reichert, Y. Nara, and M. Bleicher, J. Phys. G **52**, 035103 (2025).
[25] D. Oliinychenko, A. Sorensen, V. Koch, and L. McLerran, Phys. Rev. C **108**, 034908 (2023).
[26] J. Mohs, S. Spies, and H. Elfner, arXiv:2409.16927.
[27] Y. Nara and A. Ohnishi, Phys. Rev. C **105**, 014911 (2022).
[28] J. Hofmann, H. Stoecker, U. W. Heinz, W. Scheid, and W. Greiner, Phys. Rev. Lett. **36**, 88 (1976).
[29] H. Stöcker, Nucl. Phys. A **750**, 121 (2005).
[30] M. A. Stephanov, K. Rajagopal, and E. V. Shuryak, Phys. Rev. Lett. **81**, 4816 (1998).
[31] Y. Hatta and M. A. Stephanov, Phys. Rev. Lett. **91**, 102003 (2003); [Erratum: **91**, 129901 (2003)].
[32] T. D. Lee, *A Possible New Form of Matter at High Density*, Report of the Workshop on GeV/nucleon Collisions of Heavy Ions How and Why (Brookhaven National Lab., Upton, N.Y., USA, 1973), pp. 1–12.
[33] J. Hofmann, H. Stöcker, W. Scheid, and W. Greiner, *On the Possibility of Nuclear Shock Waves in Relativistic Heavy-Ion Collisions*, Report of the Workshop on GeV/nucleon Collisions of Heavy Ions How and Why (Brookhaven National Lab., Upton, N.Y., USA, 1973), pp. 39–47.
[34] J. E. Bernhard, J. S. Moreland, S. A. Bass, J. Liu, and U. Heinz, Phys. Rev. C **94**, 024907 (2016).
[35] D. Everett *et al.* (JETSCAPE Collaboration), Phys. Rev. C **103**, 054904 (2021).
[36] S. Pratt, E. Sangaline, P. Sorensen, and H. Wang, Phys. Rev. Lett. **114**, 202301 (2015).
[37] G. Nijs, W. van der Schee, U. Gürsoy, and R. Snellings, Phys. Rev. C **103**, 054909 (2021).
[38] M. O. Kuttan, J. Steinheimer, K. Zhou, and H. Stoecker, Phys. Rev. Lett. **131**, 202303 (2023).
[39] S. Huth *et al.*, Nature (London) **606**, 276 (2022).
[40] Y.-L. Cheng, S. Shi, Y.-G. Ma, H. Stöcker, and K. Zhou, Phys. Rev. C **107**, 064909 (2023).

[41] K. Zhou, L. Wang, L.-G. Pang, and S. Shi, Prog. Part. Nucl. Phys. **135**, 104084 (2024).
[42] M. O. Kuttan, J. Steinheimer, K. Zhou, A. Redelbach, and H. Stoecker, Phys. Lett. B **811**, 135872 (2020).
[43] M. O. Kuttan, K. Zhou, J. Steinheimer, A. Redelbach, and H. Stoecker, J. High Energy Phys. 10 (2020) 184.
[44] M. O. Kuttan, J. Steinheimer, K. Zhou, A. Redelbach, and H. Stoecker, Proc. Sci. **FAIRness2022**, 040 (2023).
[45] L.-G. Pang, K. Zhou, N. Su, H. Petersen, H. Stöcker, and X.-N. Wang, Nat. Commun. **9**, 210 (2018).
[46] Y.-L. Du, K. Zhou, J. Steinheimer, L.-G. Pang, A. Motornenko, H.-S. Zong, X.-N. Wang, and H. Stöcker, Eur. Phys. J. C **80**, 516 (2020).
[47] Y.-L. Du, D. Pablos, and K. Tywoniuk, J. High Energy Phys. 03 (2020) 206.
[48] W.-B. He, Y.-G. Ma, L.-G. Pang, H.-C. Song, and K. Zhou, Nucl. Sci. Tech. **34**, 88 (2023).
[49] F. Li, Y. Wang, H. Lü, P. Li, Q. Li, and F. Liu, J. Phys. G **47**, 115104 (2020).
[50] N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys. Rev. D **105**, 114022 (2022).
[51] Y.-D. Song, R. Wang, Y.-G. Ma, X.-G. Deng, and H.-L. Liu, Phys. Lett. B **814**, 136084 (2021).
[52] H. Hirvonen, K. J. Eskola, and H. Niemi, Phys. Rev. C **108**, 034905 (2023).
[53] F. Sergeev, E. Bratkovskaya, I. Kisel, and I. Vassiliev, Int. J. Mod. Phys. A **35**, 2043002 (2020).
[54] L. Jiang, L. Wang, and K. Zhou, Phys. Rev. D **103**, 116023 (2021).
[55] P. Murali, S. Dash, and B. K. Nandi, arXiv:2411.11001.
[56] G. Aarts, K. Fukushima, T. Hatsuda, A. Ipp, S. Shi, L. Wang, and K. Zhou, Nat. Rev. Phys. **7**, 154 (2025).
[57] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker, Phys. Rev. D **100**, 011501(R) (2019).
[58] K. Cranmer, G. Kanwar, S. Racanière, D. J. Rezende, and P. E. Shanahan, Nat. Rev. Phys. **5**, 526 (2023).
[59] J.-A. Sun, L. Yan, C. Gale, and S. Jeon, arXiv:2410.13069.
[60] H. Huang, B. Xiao, Z. Liu, Z. Wu, Y. Mu, and H. Song, Phys. Rev. Res. **3**, 023256 (2021).
[61] D. Torbunov, Y. Huang, M. Lin, Y. Ren, Y. Go, T. Rinn, H. Yu, B. Viren, and J. Huang, Phys. Rev. C **110**, 034912 (2024).
[62] M. Omana Kuttan, K. Zhou, J. Steinheimer, and H. Stöcker, companion paper, Phys. Rev. C **112**, L051902 (2025).
[63] L. V. Bravina *et al.*, Phys. Rev. C **78**, 014907 (2008).
[64] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, in *Proceedings of the 35th International Conference on Machine Learning*, edited by J. Dy and A. Krause, Proceedings of Machine Learning Research (PMLR, 2018), Vol. 80, pp. 40–49.
[65] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2019).
[66] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 4338 (2020).
[67] S. Luo and W. Hu, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York, 2021), pp. 2837–2845.
[68] Y. Sun, Y. Wang, Z. Liu, J. Siegel, and S. Sarma, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (IEEE, New York, 2020), pp. 61–70.
[69] V. Mikuni and B. Nachman, Phys. Rev. D **106**, 092009 (2022).
[70] O. Amram and K. Pedro, Phys. Rev. D **108**, 072014 (2023).
[71] F. T. Acosta, V. Mikuni, B. Nachman, M. Arratia, B. Karki, R. Milton, P. Karande, and A. Angerami, J. Instrum. **19**, P05003 (2024).
[72] V. Mikuni and B. Nachman, J. Instrum. **19**, P02001 (2024).
[73] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, G. Kasieczka, A. Korol, W. Korcari, K. Krüger, and P. McKeown, J. Instrum. **18**, P11025 (2023).
[74] A. Butter, N. Huetsch, S. Palacios Schweitzer, T. Plehn, P. Sorrenson, and J. Spinner, SciPost Phys. Core **8**, 026 (2025).
[75] M. Leigh, D. Sengupta, J. A. Raine, G. Quétant, and T. Golling, Phys. Rev. D **109**, 012010 (2024).
[76] E. Buhmann, C. Ewen, D. A. Faroughy, T. Golling, G. Kasieczka, M. Leigh, G. Quétant, J. A. Raine, D. Sengupta, and D. Shih, arXiv:2310.00049.
[77] E. Buhmann, G. Kasieczka, and J. Thaler, SciPost Phys. **15**, 130 (2023).
[78] V. Mikuni, B. Nachman, and M. Pettee, Phys. Rev. D **108**, 036025 (2023).
[79] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch, and T. Golling, SciPost Phys. **16**, 018 (2024).
[80] J. Y. Araz, V. Mikuni, F. Ringer, N. Sato, F. T. Acosta, and R. Whitehill, arXiv:2410.22421.
[81] P. Devlin, J.-W. Qiu, F. Ringer, and N. Sato, Phys. Rev. D **110**, 016030 (2024).
[82] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, New York, 2017), pp. 652–660.
[83] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv:1605.08803.
[84] J. Ho, A. Jain, and P. Abbeel, Advances in neural information processing systems **33**, 6840 (2020).
[85] M. Omana Kuttan, J. Steinheimer, K. Zhou, A. Redelbach, and H. Stoecker, Particles **4**, 47 (2021).
[86] S. Guo, H.-S. Wang, K. Zhou, and G.-L. Ma, Phys. Rev. C **110**, 024910 (2024).
[87] D. P. Kingma and M. Welling, arXiv:1312.6114.
[88] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, J. Mach. Learn. Res. **22**, 1 (2021).
[89] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, IEEE Trans. Pattern Anal. Mach. Intell. **45**, 10850 (2023).
[90] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, ACM Comput. Surv. **55**, 1 (2023).
[91] L. Wang, G. Aarts, and K. Zhou, arXiv:2311.03578.
[92] L. Wang, G. Aarts, and K. Zhou, J. High Energy Phys. 05 (2024) 060.
[93] C. Li, A. Kies, K. Zhou, M. Schlott, O. E. Sayed, M. Bilousova, and H. Stöcker, Appl. Energy **359**, 122779 (2024).